

## Remarks

Claims 1-4, 6-11, 13, 16 and 41-44 have been amended. Upon entry of the foregoing amendments, claims 1-16, and 41-44 are pending. No new matter is added by these amendments. Support for the amendments may be found in the original claims and throughout the specification, *e.g.*, at page 17, lines 17-20; page 18, lines 11-15; page 21, line 1 through page 26, line 26; page 46, line 1 through page 48, line 10; and Example 2.

Applicant thanks the Examiner for returning a copy of the initialed Form 1449, which was submitted with Applicant's Reply to Office Action filed on February 27, 2003.

### *I. Rejections under 35 U.S.C. § 112, First Paragraph (Enablement)*

Claims 1-16 and 41-44 stand rejected under 35 U.S.C. § 112, first paragraph, as containing subject matter that was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Office Action at page 2. The Office alleges that "[w]hile the specification provides some guidance for a method of determining a probability value for the above listing using the particular equations or values disclosed, the specification does not provide guidance for a method of determining probability by any other means." Office Action at page 3. The office further alleges that "[g]iven the lack of descriptive working examples in the specification, and the unpredictability of generating probability values, the specification as filed is not enabling for any method of determining the listed probability values as claimed. The instant application is only enabled for the above-mentioned computational means of the four probabilities." Office Action at page 3. Applicant respectfully disagrees.

Applicant thanks the Examiner for acknowledging that the specification is enabling for the following equations: initial oligonucleotide probability (p. 21, equation I), transition probability (p. 22, equation II), nucleic acid sequence probability (p. 23, equation III), and probability of each state for the nucleic acid sequence (p. 24, equation IV). Office Action at page 3. Applicant respectfully disagrees, however, with the

Office's allegation that the specification does not enable a person skilled in the art to practice the invention commensurate in scope with the claims.

Disclosure of a single species provides sufficient enabling support if one of skill in the art can, using the state of the art and Applicant's written disclosures, practice the invention in its full scope without undue experimentation.<sup>1</sup> See *In re Wands*, 858 F.2d 731, 737, 8 U.S.P.Q.2d 1400, 1404 (Fed. Cir. 1988); *John Hopkins Univ. v. Cellpro, Inc.*, 152 F.3d 1342, 1361, 47 U.S.P.Q.2d 1705, 1719 (Fed. Cir. 1998) (Applicant's specification provided sufficient enabling support for the Applicant's claim to immunoassay methods using a generic class of antibodies even though Applicant made a public deposit of only a single hybridoma cell line that secreted a specific antibody); *Spectra-Physics, Inc. v. Coherent, Inc.*, 827 F.2d 1524, 1533, 3 U.S.P.Q.2d 1737, 1743 (Fed. Cir. 1987), *cert. denied*, 484 U.S. 954 (1987). Section 2164.03 of the M.P.E.P. states that "[a] single embodiment may provide broad enablement in cases involving predictable factors,<sup>2</sup> such as mechanical or electrical elements." Citing *In re Vickers*, 141 F.2d 522, 526-27, 61 U.S.P.Q. 122, 127 (C.C.P.A. 1944); *In re Cook*, 439 F.2d 730, 734, 169 U.S.P.Q. 298, 301 (C.C.P.A. 1971). Furthermore, it is well established law that patent applicants are not required to disclose every species enabled by their claims. See *In re Vaeck*, 947 F.2d 488, 496, 20 U.S.P.Q.2d 1438, 1445 (Fed. Cir. 1991).

Applicant need only show that one skilled in the art would be able to make and use the claimed invention using the application as a guide. *In re Brandstadter*, 484 F.2d 1395, 1406-07, 179 U.S.P.Q. 286, 294 (C.C.P.A. 1973). In order to be enabling, the

---

<sup>1</sup> Applicant notes that the performance of routine and well-known steps cannot create undue experimentation even if it is laborious. See *In re Wands*, 858 F.2d at 737, 8 U.S.P.Q.2d at 1404; *In re Angstadt*, 537 F.2d 498, 504, 190 U.S.P.Q. 214, 218-219 (C.C.P.A. 1976). Time and difficulty of experiments are not determinative if they are merely routine. M.P.E.P. § 2164.06, page 2100-186.

<sup>2</sup> Applicant respectfully disagrees with the Office's implied assertion that determining probabilities using pre-existing statistical methods is unpredictable. The Office states, "[g]iven the lack of descriptive working examples in the specification, and the unpredictability of generating probability values, the specification as filed is not enabling for any method of determining the listed probability values as claimed." Office Action at page 3 (italics added). Applicant respectfully submits that determining probability values using pre-existing statistical methods (*i.e.*, known mathematical equations) is not unpredictable. Applicant respectfully requests that the Office provide legal or other support for the assertion that generating probability values is "unpredictable." Office Action at page 3.

specification need not disclose what is well-known to those skilled in the art and preferably omits that which is well known to those skilled and already available to the public.<sup>3</sup> See, e.g., M.P.E.P. § 2164.05(a), page 2100-185, citing *In re Buchner*, 929 F.2d 660, 661, 18 U.S.P.Q. 2d 1331, 1332 (Fed. Cir. 1991); *Hybritech, Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 1384, 231 U.S.P.Q. 81, 94 (Fed. Cir. 1986), cert. denied, 480 U.S. 947 (1987); and *Lindemann Maschinenfabrik GMBH v. American Hoist and Derrick Co.*, 730 F.2d 1452, 1463, 221 U.S.P.Q. 481, 489 (Fed. Cir. 1984).

Applicant respectfully submits that the specification as filed is enabling for the full scope of the claims. The specification describes, and provides working examples for, the use of inhomogeneous Markov models to determine the probabilities for each of the one or more states for a selected nucleotide. See, e.g., specification at pages 19, line 7 though page 27, line 6, and Examples 1 through 3. As such, specification provides sufficient support to enable one of skill in the art, using the state of the art and the specification disclosure, to practice the invention in its full scope without undue experimentation.

Moreover, although Applicant respectfully maintains that no additional information is needed to enable the full scope of the claims, the specification also provides that "[a]ny probability model applicable to nucleic acid sequence state probabilities can be used for the probability steps if the output of the probability model sufficiently supports the method, including inhomogeneous Markov models having fewer than eight states." See specification at page 19, lines 22-24. The specification also points that skilled artisan to Durbin *et al.*, *Biological Sequence Analysis* (1998), described at page 19, lines 26-27 of Applicant's disclosure.<sup>4</sup> Applicant respectfully asserts that for at least these reasons, the specification as filed provides adequate guidance to enable one of

---

<sup>3</sup> Applicant respectfully submits that the Office has failed to provide any evidence to suggest that the statistical methods taught by Durbin are not well known in the art. Applicant points the Office to the Bibliography of Durbin, which cites over 200 references written by a diversity of authors. Durbin, pages 326-344.

<sup>4</sup> Copies of the Bibliography of Durbin, as well as Durbin Chapters 5 and 11, are enclosed for the Examiner's convenience. See Exhibit A.

skill in the art to practice the invention using additional statistical methods that would be substitutable for the four equations that the Office has determined to be enabled.

Moreover, Applicant respectfully disagrees with the Office's implied assertion that the material referred to in Durbin is "essential material." Office Action at page 4; and Office Action mailed January 13, 2003 at page 5. The M.P.E.P. defines "essential material" as including "that which is necessary to provide an enabling disclosure of the claimed invention." M.P.E.P. § 608.01(p), page 600-79.

Applicant respectfully submits that the material in Durbin is not "essential" because it is not necessary to provide an enabling disclosure of the claimed invention. As stated above, disclosure of a single species provides sufficient enabling support if one of skill in the art can, using the state of the art and Applicant's written disclosures, practice the invention in its full scope without undue experimentation. *See In re Wands*, 858 F.2d at 737; *John Hopkins Univ.*, 152 F.3d at 1361; *Spectra-Physics, Inc.*, 827 F.2d at 1533; M.P.E.P. § 2164.03. Furthermore, it is well established law that patent applicants are not required to disclose every species enabled by their claims. *See In re Vaeck*, 947 F.2d at 496.

The Office alleges that "Applicant's reliance on prior art methods may only extend to well known methods and that single specific publications do not support their content as being well known." Office Action at pages 4-5. Applicant disagrees. Applicant reiterates that the Office has not provided evidence to suggest that the methods of Durbin are not well known in the art. *See footnote 3 infra*. Applicant respectfully submits that the Office has offered no legal support for the assertion that "single specific publications do not support their content as being well known." Furthermore, Applicant's citation of a single reference, rather than a list of references, cannot properly be used as evidence that the information contained therein is not well known in the art. After all, requiring patent applicants to cite a list of cumulative references would contravene the well-known principle that the specification need not disclose what is well-known to those skilled in the art and preferably omits that which is well known to those skilled and already available to the public. M.P.E.P. § 2164.05(a), page 2100-185.



For at least the foregoing reasons, Applicant respectfully asserts that the specification as filed enables a person of skill in the art to practice the invention commensurate in scope with the claims. Applicant respectfully submits that the rejection of claims 1-16 and 41-44 under 35 U.S.C. § 112, first paragraph is improper and should be withdrawn. Reconsideration and withdrawal of these rejections are respectfully requested.

Should the Examiner maintain this rejection based on the contention that the material disclosed in Durbin is not well known to those of ordinary skill in the art, Applicant respectfully requests that the Examiner support this contention by way of affidavit in accordance with 37 C.F.R. § 1.104 (d)(2).

***II. Rejections under 35 U.S.C. § 112, Second Paragraph (Indefiniteness)***

Claims 1-16 and 41-44 stand rejected under 35 U.S.C. § 112, second paragraph as being allegedly indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Office Action at page 5.

***(a) Rejection of claims 3 and 11***

Claims 3 and 11 stand rejected under 35 U.S.C. § 112, second paragraph on the grounds that they contain mathematical equations that are allegedly confusing as they incorporate “ $\Phi(f)$ ” representing bias which cancels itself out in each equation, and therefore nullifies its effect on the equation.” Office Action at page 5. The Office further alleges that “[i]f the Applicant intends this bias not to be represented [sic] by the same exact number in the numerator and denominator, then subscripts, or some other form of notation, would be needed in order to clarify this issue.” Office Action at page 5. Applicant respectfully disagrees.

Applicant respectfully disagrees that “ $\Phi(f)$ ” (representing bias) cancels itself out of the equation. Applicant respectfully points out that “ $\Phi(f)$ ” corresponds to a function, and as such, “ $\Phi(f)$ ” can have different numerical values corresponding to different elements in the set of states. *See, e.g.*, specification at page 47, lines 13-20. As acknowledged by the Examiner, when “ $\Phi(f)$ ” has different numerical values

corresponding to different elements in the set of states, " $\Phi(f)$ " has different values in the numerator and denominator of the equations in claims 3 and 11, and hence " $\Phi(f)$ " does not cancel out. *Compare, e.g.*, calculation at page 46, lines 1-5 with calculation on page 48, lines 5-10. Applicant therefore disagrees that bias cancels itself out of the equation.

Applicant also respectfully disagrees with the Office's assertion that subscripts or other notation are required to clarify this issue. Applicant respectfully submits that acceptability of the claim language depends on whether one of ordinary skill in the art would understand what is claimed, in light of the specification. M.P.E.P. § 2173.05(b). Applicant points out that the specification clearly defines " $\Phi(f)$ " as a function. *See, e.g.*, specification at page 24, lines 4-5 and lines 18-25. Applicant respectfully submits that one of skill in the art would understand that a function may be assigned different values under different circumstances, and would also understand that Example 2 illustrates that the values substituted for " $\Phi(f)$ " do not cancel out of the equation. *Compare, e.g.*, calculation on page 46, lines 1-5, with calculation on page 48, lines 5-10. Applicant therefore respectfully submits that one of skill in the art would understand the meaning of " $\Phi(f)$ " in light of the specification, and that no subscripts or notations are necessary to clarify the issue.

For the foregoing reasons, Applicant respectfully asserts that the specification contains guidelines sufficient to teach the meaning of the claim language " $\Phi(f)$ " to one of ordinary skill in the art, and thus, the rejection of claims 3 and 11 under 35 U.S.C. § 112, second paragraph is improper and should be withdrawn. Reconsideration and withdrawal of this rejection is respectfully requested.

**(b) Rejection of Claims 1, 7, 8, and 41-44**

Claim 1 stands rejected under 35 U.S.C. § 112, second paragraph, on the grounds that it recites the phrase "said probability of said nucleic acid sequence" which is allegedly "vague and indefinite due to the lack of clear antecedent basis for the noted phrase in part d) of claim 1." Office Action at page 5. The Office further alleges that "[t]his lack of antecedent basis and unclear wording is also present in other independent claims 7, 8 (regarding part d) said window probability), 41, 42 (part a) probability of a

window), 43, and 44 (part d) said window probability). This rejection is also applicable to claims 2-6 and 9-16 which are claims dependent from said independent claims due to their direct or indirect dependence." Office Action at page 6. Applicant respectfully disagrees.

Applicant disagrees that there is a lack of clear antecedent basis for the phrase "said probability of said nucleic acid sequence." However, in order to facilitate prosecution, Applicant has amended claim 1.

Applicant further disagrees that there is a lack of antecedent basis and unclear wording in claims 7, 8, and 41-44. Applicant respectfully submits that the specification defines "window" as "a contiguous and defined number of nucleotides within a nucleic acid sequence." *See, e.g.*, Specification at page 17, lines 17-20. Applicant also directs the Office to page 25, line 25-26 of the specification, which states "[i]n order to determine the state probabilities for more than one nucleotide, a window is used for each nucleotide that is examined." Applicant therefore submits that one of ordinary skill, reading the claims in light of the specification and in light of his or her knowledge of the art, would understand the meaning of the phrase "said window probability." When read in light of the specification, the phrase "said window probability" is no less understandable than the phrase "said initial oligonucleotide probability." However, in order to facilitate prosecution, Applicant has amended claims 7, 8, 41, and 44.

Applicant therefore submits that the grounds for the rejection of Claim 1, 7, 8, and 41-44 has been rendered moot. Applicant further submits that the amendments to claims 1, 8, and 41-44 has also rendered moot the rejections of dependent claims 2-6 and 9-16. In light of these remarks, Applicant respectfully requests withdrawal of these rejections.

**(c) Rejection of Claims 1, 7, 8, and 41-44**

Claims 1, 7, 8, and 41-44 stand rejected under 35 U.S.C. § 112, second paragraph, on the grounds that they recites the phrase "based upon" which allegedly renders unclear "the metes and bounds of the parameters that that determine how much basis is included upon the determinations." Office Action at page 6. The Office further alleges that

"[c]laims 2-6 and 9-16 are also indefinite due to their dependency from claims 1 and 8."  
Office Action at page 6. Applicant respectfully disagrees.

Applicant disagrees that the phrase "based upon" renders unclear the metes and bounds of the claim. However, in order to facilitate prosecution, Applicant has amended claims 1, 7, 8, and 41-44.

Applicant therefore submits that the grounds for the rejection of Claim 1, 7, 8, and 41-44 has been rendered moot. Applicant further submits that the amendments to claims 1, 8, and 41-44 has also rendered moot the rejections of dependent claims 2-6 and 9-16. In light of these remarks, Applicant respectfully requests withdrawal of these rejections.

**(d) Rejection of Claim 7**

Claim 7 stands rejected under 35 U.S.C. § 112, second paragraph, on the grounds that it recites the term "capable", which allegedly is a relative term that renders the claim indefinite. Applicant respectfully disagrees.

Even if "capable" were a relative term, the use of a relative term does not make a claim *per se* indefinite. *Seattle Box Co. v. Industrial Crating & Packing, Inc.*, 731 F.2d 818, 826, 221 U.S.P.Q. 568, 574 (Fed. Cir. 1984); M.P.E.P. § 2173.05(b). Breadth in a claim is not to be equated with indefiniteness. *In re Miller*, 441 F.2d 689, 169 U.S.P.Q. 597 (C.C.P.A 1971); M.P.E.P. § 2173.04. The words of a claim must be given their plain meaning unless they are defined in the specification. M.P.E.P. § 2111.01, page 2100-47.

For at least these reasons, Applicant submits that Claim 7 is not indefinite in the recitation of "capable." However, in order to facilitate prosecution, Applicant has amended Claim 7. Applicant therefore submits that the grounds for the rejection of Claim 7 has been rendered moot. In light of these remarks, Applicant respectfully requests withdrawal of this rejection.

**(e) Rejection of Claims 3 and 11**

Claims 3 and 11 stand rejected under 35 U.S.C. § 112, second paragraph, on the grounds that they are allegedly "vague and indefinite due to the lack of clarity in the

following terms: **f**, **S**, **P<sub>f</sub>**, **P<sub>i</sub>**, and **Φ**." Office Action at page 7. The Office further alleges that "a clarification of the metes and bounds is required, by listing in the claim the exact definition of each term in order to make clear whether definitions from the art should be utilized or those in the specification since, as argued by Applicant, art defined (not specification defined) methods are apparently heavily relied upon by Applicant." Office Action at page 7. Applicant respectfully disagrees.

The test for determining whether terms in a given claim are indefinite is whether one skilled in the art would understand what is claimed. *Amgen, Inc. v. Chugai Pharmaceutical Co., Ltd.*, 927 F.2d 1200, 18 U.S.P.Q.2d 1016 (Fed. Cir. 1991), *cert denied*, 112 S.Ct. 169 (1991). M.P.E.P. § 2173.02 states that "[d]efiniteness of claim language must be analyzed, not in a vacuum, but in light of: (A) The content of the particular application disclosure; (B) The teachings of the prior art; and (C) The claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made."

Under M.P.E.P. § 2173.02, the meaning of the terms **f**, **S**, **P<sub>f</sub>**, **P<sub>i</sub>**, and **Φ** must be determined in light of factors (A) through (C) listed above. Applicant respectfully submits that one of skill in the art would understand that **f**, **S**, **P<sub>f</sub>**, **P<sub>i</sub>**, and **Φ** correspond to terms, or parts of terms, of a mathematical equation. Applicant further directs the Office to pages 21-25 of the specification, and Examples 1-2. Applicant respectfully points out that they know of no legal requirement to list "the exact definition of each term" within the claim, and respectfully requests that the Examiner state the legal basis which is relied upon for this statement.

For at least these reasons, Applicant submits that one of ordinary skill in the art, when reading the claim terms **f**, **S**, **P<sub>f</sub>**, **P<sub>i</sub>**, and **Φ** in light of the specification and the teachings of the prior art, would understand what was meant by Claims 3 and 11. Therefore, Applicant respectfully requests that the indefiniteness rejections of claim 3 and 11 under 35 U.S.C. § 112, second paragraph, be withdrawn.

*(f) Rejection of Claims 8 and 44*

Claims 8 and 44 stand rejected under 35 U.S.C. § 112, second paragraph, on the grounds that they allegedly lack clarity due to the claim language “determining a probability for said window for each of said states. Claims 9-16 are also indefinite due to their dependency from claim 8.” Office Action at page 7. Applicant respectfully disagrees.

The Office alleges that “a probability cannot be determined for a window, but rather the states found in the window.” Office Action at page 7. Applicant respectfully disagrees. As stated above, under M.P.E.P. § 2173.02, the meaning of the phrase “determining a probability of said window for each of said states” must be determined in light of (A) The content of the particular application disclosure; (B) The teachings of the prior art; and (C) The claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made. Applicant respectfully submits that the specification defines “window” as “a contiguous and defined number of nucleotides within a nucleic acid sequence.” *See, e.g.*, Specification at page 17, line 17. Applicant therefore submits that one of ordinary skill, reading this phrase in light of the specification and his or her knowledge of the art, would understand the meaning of the phrase “determining a probability of said window for each of said states.” When read in light of the specification, the phrase “determining a probability of said window for each of said states” is no less understandable than the phrase “determining an initial oligonucleotide probability.” However, in order to facilitate prosecution, claims 8 and 44 have been amended.

Applicant respectfully submits that, in light of the above arguments, the grounds for the rejection of Claims 8 and 44 has been overcome or rendered moot. Applicant further submits that the rejections of dependent claims 9-16 has also been overcome or rendered moot. In light of these remarks, Applicant respectfully requests withdrawal of these rejections.

**III. Rejections under 35 U.S.C. § 102(b)**

Claims 1, 4, 5, 7-9, 12, 13, 15, and 41-44 stand rejected under 35 U.S.C. § 102(b) as being allegedly anticipated by Borodovsky *et al.* (Computers Chem., 1993). The Office alleges that "Due to the confusion (see 35 U.S.C. 112, 2<sup>nd</sup> paragraph rejection above) of " $\Phi(f)$ " effectively canceling itself out in the equations of claims 3 and 11, these equations are equivalent to the equations listed on page 129 (Borodovsky *et al.*). Being equivalent equations, if one probability (as provided by Applicant) is "capable of accepting a bias" (claim 7, line 10), then the same probability stated by Borodovsky *et al.* (page 129) must also be capable of accepting a bias. Therefore, Borodovsky *et al.* anticipate the instant invention." Applicant respectfully disagrees.

As noted above, Applicant respectfully disagrees that " $\Phi(f)$ " (representing bias) cancels itself out of the equation. Applicant respectfully points out that " $\Phi(f)$ " corresponds to a function, and as such, " $\Phi(f)$ " can have different numerical values corresponding to different elements in the set of states. *See, e.g.*, specification at page 47, lines 13-20. As acknowledged by the Examiner, when " $\Phi(f)$ " has different numerical values corresponding to different elements in the set of states, " $\Phi(f)$ " has different values in the numerator and denominator of the equations in claims 3 and 11, and hence " $\Phi(f)$ " does not cancel out. For example, Applicant points the Office to Example 2, pages 46-48 of the specification, which illustrates that the values substituted for " $\Phi(f)$ " do not cancel out of the equation. *Compare, e.g.*, calculation at page 46, lines 1-5 with calculation on page 48, lines 5-10. Applicant therefore disagrees that bias cancels itself out of the equation.

Applicant respectfully disagrees that claims 1, 4, 5, 7-9, 12, 13, 15, and 41-44 are anticipated under §102(b) by Borodovsky. As noted by the Examiner, anticipation under §102(b) requires that every element of a claim appears in a single reference. Applicant respectfully asserts that claims 1, 4, 5, 7-9, 12, 13, 15, and 41-44 each contain the function " $\Phi(f)$ " (or the phrase "bias function"), which is lacking in Borodovsky. Applicant respectfully submits that claims 1, 4, 5, 7-9, 12, 13, 15, and 41-44, as amended herein, are therefore not anticipated by Borodovsky. Applicant therefore respectfully requests withdrawal of the rejections under 35 U.S.C. §102(b).

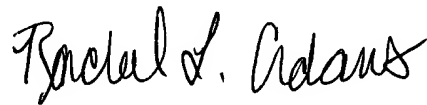
In addition, the Office alleges that "if one claim is 'capable of accepting a bias' (claim 7, line 10), then the same probability stated by Borodovsky *et al.* (page 129) must also be capable of accepting a bias." Office Action at page 8. Applicant respectfully submits that claim 7 has been amended, and that, in light of the amendment of claim 7, the grounds for the rejection of claim 7 have been overcome or rendered moot. In light of these remarks, Applicant respectfully requests withdrawal of the rejection of claim 7.



### Conclusion

In view of the above, each of the presently pending claims is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to withdraw the outstanding rejections of the claims and to pass this application to issue. The Examiner is encouraged to contact the undersigned at (202) 942-5512 should any additional information be necessary for allowance.

Respectfully submitted,



Rachel L. Adams (Reg. Attorney No. 54,660)  
David R. Marsh (Reg. Attorney No. 41,408)  
Holly Logue Prutz (Reg. Attorney No. 47,755)

Date: August 12, 2003

ARNOLD & PORTER  
555 Twelfth Street, N.W.  
Washington, D.C. 20004-1206  
(202) 942-5000 telephone  
(202) 942-5999 facsimile

## Profile HMMs for sequence families

So far we have concentrated on the intrinsic properties of single sequences, such as CpG islands in DNA, or on pairwise alignment of sequences. However, functional biological sequences typically come in families, and many of the most powerful sequence analysis methods are based on identifying the relationship of an individual sequence to a sequence family. Sequences in a family will have diverged from each other in their primary sequence during evolution, having separated either by a duplication in the genome, or by speciation giving rise to corresponding sequences in related organisms. In either case they normally maintain the same or a related function. Therefore, identifying that a sequence belongs to a family, and aligning it to the other members, often allows inferences about its function.

If you already have a set of sequences belonging to a family, you can perform a database search for more members using pairwise alignment with one of the known family members as the query sequence. To be more thorough, you could even search with all the known members one by one. However, pairwise searching with any one of the members may not find sequences distantly related to the ones you have already. An alternative approach is to use statistical features of the whole set of sequences in the search. Similarly, even when family membership is clear, accurate alignment can be often be improved significantly by concentrating on features that are conserved in the whole family.

How, in brief, do we identify such features? Just as a pairwise alignment captures much of the relationship between two sequences, a multiple alignment can show how the sequences in a family relate to each other. Figure 5.1 shows a multiple alignment of seven sequences from the large globin family (hundreds of globin sequences are available in the protein sequence databases). The three dimensional structure has been obtained for each protein in the alignment shown, and the sequences have been aligned on the basis of aligning the eight alpha helices of the conserved globin fold, and also on the basis of aligning certain key residues in the sequences, such as two conserved histidines (H) which are the residues which interact with an oxygen-binding heme prosthetic group in the globin active site.

It is clear that some positions in the globin alignment are more conserved than others. In general the helices are more conserved than the loop regions between

Helix  
HBA\_HUI  
HBB\_HUI  
MYG\_PH  
GLB3\_CI  
GLB5\_PI  
LGB2\_LI  
GLB1\_GI  
Consens

Helix  
HBA\_HUN  
HBB\_HUN  
MYG\_PHY  
GLB3\_CI  
GLB5\_PI  
LGB2\_LI  
GLB1\_GI  
Consens

Helix  
HBA\_HUN  
HBB\_HUN  
MYG\_PHY  
GLB3\_CI  
GLB5\_PI  
LGB2\_LI  
GLB1\_GI  
Consens

Fi  
Le  
da  
A-  
res  
ca  
wh

them, ar  
a new se  
these m  
tion wil

As m  
a proba  
den Ma  
profile I  
structur  
& Eisen  
hidden I

We w  
multiple  
and sco

```

Helix      AAAAAAAAAAAAAAAAAA  BBBBBBBBBBBBBBBBBBCCCCCCCCCCCC
HBA_HUMAN  -----VLSPADKTNVKAAGKVGAA--HAGEYGAEALERMFLSPPTTKTYFPHF
HBB_HUMAN  -----VHLTPEEKSAVTALWGKV----NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  -----VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRLF
GLB3_CHITP -----LSADQISTVQASFDKVGK-----DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSVAPLSAAEKKIRSAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU -----GALTESQAALVKSSWEEFN--NIPKHTHRFFILVLEIAPAAKDLFS-F
GLB1_GLYDI -----GLSAAQRQVIAATWKDIAGADNGAGVGKDLIKFLSAHPQMAAVFG-F
Consensus  Ls.... v a W kv . . g . L . f . P . F F

Helix      DDDDDDDDEEEEEEEEEEEEEEEEEEEEEEE FFFFFFFF
HBA_HUMAN  -DLS-----HGSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL-
HBB_HUMAN  GDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHL---D--NLKGTFTATLSELHCDKL-
MYG_PHYCA  KHLKTEAEAMKASEDLKKHGVTVLTALGAILKK---K-GHHEAELKPLAQSHATKH-
GLB3_CHITP AG-KDLESIKGTAPFETHANRIVGFFSKIIGEL--P---NIEADVNTFVASHKPRG-
GLB5_PETMA KGLTTADQLKKSADVRWHAERIINAVNDAVASM--DDTEKMSMKLRDLSGKHAKSF-
LGB2_LUPLU LK-GTSEVPQNNPELQAHAGKVFKLVEAAIQLVTVGVVTDATLKNLGSVHVSKG-
GLB1_GLYDI SG---AS---DPGVAALGAKVLAQIGVAVSHL--GDEGKMVAQMKAVGRHKGYGN
Consensus  . t . . . v..Hg kv. a a..l d . a l l H .

Helix      FFGGGGGGGGGGGGGGGGGGGG HHHHHHHHHHHHHHHHHHHHHHHHHHH
HBA_HUMAN  -RVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR-----
HBB_HUMAN  -HVDPENFRLLGNVLVLCVLAHFGKEFTPPVQAAYQKVAGVANALAHKYH-----
MYG_PHYCA  -KIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
GLB3_CHITP -VTHDQLNNFRAGFVSYMKAHT--DFA-GAEAAGCATLDTFFGMIFSKM-----
GLB5_PETMA -QVDPQYFKVLA AVIADTVAAG-----DAGFEKLMSMICILLRSAY-----
LGB2_LUPLU --VADAHFPVVKEAIIKTIKEVVGAKWSEELNSAWTIADELAIVIKKEMNDAA--
GLB1_GLYDI KHIAQYFEPGLGASLLSAMEHRIGGKMNAAKDAWAAAYADISGLISGLQS-----
Consensus  v . f l . . . . . f . aa. k . . l sky

```

**Figure 5.1** An alignment of seven globins from Bashford, Chothia & Lesk [1987]. To the left is the protein identifier in the SWISS-PROT database [Bairoch & Apweiler 1997]. The eight alpha helices are shown as A-H above the alignment. A consensus line below the alignment indicates residues that are identical among at least six of the seven sequences in upper case, ones identical in four or five sequences in lower case, and positions where there is a residue identical in three sequences with a dot.

them, and certain residues are particularly strongly conserved. When identifying a new sequence as a globin, it would be desirable to concentrate on checking that these more conserved features are present. How to obtain and use such information will be the subject of this chapter.

As might be expected, our approach to consensus modelling will be to make a probabilistic model. In particular, we will develop a particular type of hidden Markov model well suited to modelling multiple alignments. We call these *profile HMMs* after standard *profiles*, which are closely related non-probabilistic structures introduced previously for the same purpose by Gribskov, McLachlan & Eisenberg [1987]. Profile HMMs are probably the most popular application of hidden Markov models in molecular biology at the moment [Eddy 1996].

We will assume for the purposes of this chapter that we are given a correct multiple alignment, from which we will build a model that can be used to find and score potential matches to new sequences. The multiple alignment could

be built from structural information, like the globin alignment shown here, or it could come from a sequence-based alignment procedure, such as those discussed in Chapter 6.

Much of this chapter makes use of the theory presented in Chapter 3 for general HMMs. The most important algorithms will be presented again in the specific form relevant to profile HMMs. There is also an extensive discussion of how to estimate optimal probability parameters from multiple sequence alignments.

## 5.1 Ungapped score matrices

One general feature of protein family multiple alignments, which can be seen in Figure 5.1, is that gaps tend to line up with each other, leaving solid blocks where there are no insertions or deletions in any of the sequences. We will start by considering models for these ungapped regions.

As an example, consider the E helix of Figure 5.1. A natural probabilistic model for such a region would be to specify independent probabilities  $e_i(a)$  of observing amino acid  $a$  in position  $i$  (we use letter  $e$  because these will turn out to be the *emission probabilities* of the hidden Markov model when we introduce gaps). The probability of a new sequence  $x$  according to this model is then

$$P(x|M) = \prod_{i=1}^L e_i(x_i),$$

where  $L$  is the length of the block, 21 in this case. As usual, we are in fact more interested in the ratio of this probability to the probability of  $x$  under a random model, and so to test for membership in the family we evaluate the log-odds ratio

$$S = \sum_{i=1}^L \log \frac{e_i(x_i)}{q_{x_i}}.$$

The values  $\log \frac{e_i(a)}{q_a}$  behave like elements in a score matrix  $s(a, b)$ , where the second index is position  $i$ , rather than amino acid  $b$ . For this reason, such an approach is known as a *position specific score matrix* (PSSM). A PSSM can be used to search for a match in a longer sequence  $x$  of length  $N$  by evaluating the score  $S_j$  for each starting point  $j$  in  $x$  from 1 to  $N - L + 1$ , where  $L$  is the length of the PSSM.

## 5.2 Adding insert and delete states to obtain profile HMMs

Although a PSSM captures some conservation information, it is clearly an inadequate representation of all the information in a multiple alignment of a protein

family.  
bine the  
by Hen  
sue her  
of the a

One  
same g  
is also  
of whe  
give us  
positio

The  
itive st  
vide a  
off by  
identic  
bility 1

Alignr  
emissi

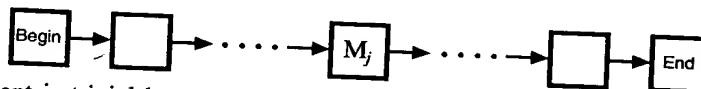
The  
arately  
the me  
inserti  
The  $I_i$   
groun  
wise  
itself,  
 $M_{i+1}$

We d  
inser  
that  
emis

family. We have to find some way to take account of gaps. It is possible to combine the scores of multiple ungapped block models, and this is the approach taken by Henikoff & Henikoff [1991] in the BLOCKS database. However, we will pursue here the aim of developing a single probabilistic model for the whole extent of the alignment.

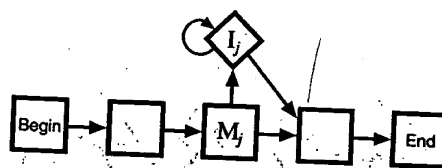
One approach is to allow gaps at each position in the alignment, using the same gap score  $\gamma(g)$  at each position, as in pairwise alignment. However, this is also ignoring information, because the alignment gives us explicit indications of where gaps are more and less likely. We want to capture this information to give us position sensitive gap scores, just as the emission probabilities gave us position sensitive substitution scores.

The approach we take is to build a hidden Markov model (HMM), with a repetitive structure of states, but different probabilities in each position. This will provide a full probabilistic model for sequences in the sequence family. We start off by observing that the PSSM can be viewed as a trivial HMM with a series of identical states that we will call *match* states, separated by transitions of probability 1.



Alignment is trivial because there is no choice of transitions. We rename the emission probabilities for the match states to  $e_{M_i}(a)$ .

The next step is to deal with gaps. We must treat insertions and deletions separately. To handle insertions, i.e. portions of  $x$  that do not match anything in the model, we introduce a set of new states  $I_i$ , where  $I_i$  will be used to match insertions after the residue matching the  $i$ th column of the multiple alignment. The  $I_i$  have emission distribution  $e_{I_i}(a)$ , but these are normally set to the background distribution  $q_a$ , just as for seeing an unaligned inserted residue in a pairwise alignment. We need transitions from  $M_i$  to  $I_i$ , a loop transition from  $I_i$  to itself, to accommodate multi-residue insertions, and a transition back from  $I_i$  to  $M_{i+1}$ . Here is a single insert state of this kind:

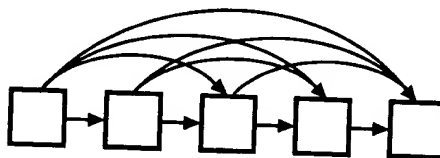


We denote insert states in our diagrams by diamonds. The log-odds cost of an insert is the sum of the costs of the relevant transitions and emissions. Assuming that  $e_{I_i}(a) = q_a$  as described above, there is no log-odds contribution from the emission, and the score of a gap of length  $k$  is

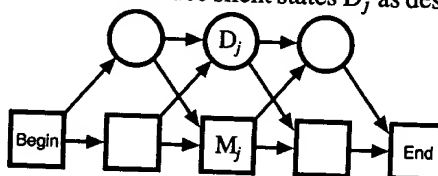
$$-\log a_{M_j I_j} + \log a_{I_j M_{j+1}} + (k-1) \log a_{I_j I_j}.$$

From this you can see that the type of insert state shown corresponds to an affine gap scoring model.

Deletions, i.e. segments of the multiple alignment that are not matched by any residue in  $x$ , could be handled by forward 'jump' transitions between non-neighbouring match states:



However, to allow arbitrarily long gaps in a long model this way would require a lot of transitions. Instead we introduce silent states  $D_j$  as described in Section 3.4:



Because the silent states do not emit any residues, it is possible to use a sequence of them to get from any match state to any later one, between two residues in the sequence. The cost of a deletion will then be the sum of the costs of an  $M \rightarrow D$  transition followed by a number of  $D \rightarrow D$  transitions, then a  $D \rightarrow M$  transition. This is at first sight exactly analogous to the cost of an insert, although the path through the model looks different. In detail, it is possible that the  $D \rightarrow D$  transitions will have different probabilities, and hence contribute differently to the score, whereas all the  $I \rightarrow I$  transitions for one insert involve the same state, and so are guaranteed to have the same cost.

The full resulting HMM has the structure shown in Figure 5.2. This form of model, which we call a profile HMM, was first introduced in Haussler *et al.* [1993] and Krogh *et al.* [1994]. We have added transitions between insert and delete states, as they did, although these are usually very improbable. Leaving them out has negligible effect on scoring a match, but can create problems when building the model.

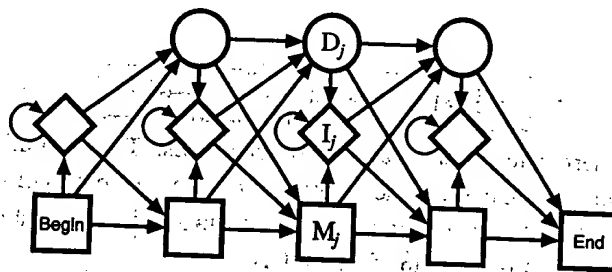


Figure 5.2 The transition structure of a profile HMM. We use diamonds to indicate the insert states and circles for the delete states.

We have used in it is us which

Let then Fi coming sequen to inca values probab probab In f tained one of finding same : scribe standa below.

5

Altho progr intro struct to cap the w sus se Th multi illust ment edit s 21 not A mc & Eisen Eisen



### Profile HMMs generalise pairwise alignment

We have seen how the costs of using gap states in a profile HMM mirror those used in pairwise alignment with affine gaps. To help make clear the relationship, it is useful to consider the degenerate case where the multiple alignment from which we build the HMM contains just one sequence.

Let us compare Figure 5.2 with Figure 4.2. If we call the example sequence  $y$ , then Figure 5.2 is an unrolled version of Figure 4.2, with the  $y_i$  emissions each coming from a separate copy of the pair HMM. The states  $M_j$  correspond to a sequence of match states  $M$ , the  $I_j$  to corresponding incarnations of  $X$ , and the  $D_j$  to incarnations of  $Y$ . To achieve as close a correspondence as possible, the natural values for the match emission probabilities  $e_{M_i}(a)$  are  $p_{y_i a}/q_{y_i}$ , the conditional probabilities of seeing  $a$  given  $y_i$  in a pairwise alignment, and for the transition probabilities  $a_{M_i I_i} = a_{M_i D_{i+1}} = \delta$  and  $a_{I_i I_i} = a_{D_i D_{i+1}} = \varepsilon$  for all  $i$ .

In formal terms our profile HMM is effectively the hidden Markov model obtained by conditioning the pair HMM of Figure 4.2 on emitting sequence  $y$  as one of the sequences in its alignment. Because of this, the Viterbi equations for finding the most probable alignment of  $x$  to our profile HMM are essentially the same as those for the most probable alignment of  $x$  and  $y$  to the pair HMM described in Chapter 4. If we convert them into log-odds ratio form we recover our standard affine gap cost pairwise alignment equations of (2.16), as we will see below. Any differences are due to slightly different Begin and End arrangements.

### 5.3 Deriving profile HMMs from multiple alignments

Although it is nice to see that the profile HMM is doing the same sort of dynamic programming as we have used before for pairwise alignment, this is not why we introduced them. The key idea behind profile HMMs is that we can use the same structure as shown in Figure 5.2, but set the transition and emission probabilities to capture specific information about each position in the multiple alignment of the whole family. Essentially, we want to build a model representing the consensus sequence for the family, not the sequence of any particular member.

There are a number of different ways to derive the parameter values from a multiple alignment of the sequences in the family. To provide an example for illustrating these methods, Figure 5.3 shows a short section of the globin alignment shown in Figure 5.1.

#### Non-probabilistic profiles

A model similar to the profile HMM was first introduced by Gribskov, McLachlan & Eisenberg [1987] who coined the name 'profile' (see also Gribskov, Lüthy & Eisenberg [1990]). However, they did not have an underlying probabilistic model,

HBA_HUMAN	...VGA--HAGEY...
HBB_HUMAN	...V----NVDEV...
MYG_PHYCA	...VEA--DVAGH...
GLB3_CHITP	...VKG-----D...
GLB5_PETMA	...VYS--TYETS...
LGB2_LUPLU	...FNA--NIPKH...
GLB1_GLYDI	...IAGADNGAGV...
	*** *****

**Figure 5.3** Ten columns from the multiple alignment of seven globin protein sequences shown in Figure 5.1. The starred columns are ones that will be treated as 'matches' in the profile HMM.

but rather directly assigned position specific scores for each match state and gap penalty, for use in standard 'best match' dynamic programming. They set the scores for each consensus position to the averages of the standard substitution scores from all the residues seen in the corresponding multiple alignment column. For example, they would set the score for residue  $a$  in column 1 of our example to be

$$\frac{5}{7}s(V, a) + \frac{1}{7}s(F, a) + \frac{1}{7}s(I, a)$$

where  $s(a, b)$  is the standard substitution matrix. They also set gap penalties for each column using a heuristic equation that decreased the cost of a gap (either insertion or deletion) according to the length of the longest gap observed in the multiple alignment spanning the column.

Although this seems an intuitively obvious way to combine information, and it has been used effectively by many people for finding new members of families, it does produce anomalies. For example, column 1 is much more strongly conserved than column 2 in the example shown in Figure 5.3, but the information in column 1 will be smeared out just as much by the substitution matrix as that in column 2. If we had an alignment with 100 sequences, all with a cysteine (C) at some position, then the implicit probability distribution for that column for an 'average' profile would be exactly the same as would be derived from a single sequence. This does not correspond to our expectation that the likelihood of a cysteine should go up as we see more confirming examples.

In addition to these observations about substitution scores, the scores for gaps do not behave as expected. For example, from the alignment in Figure 5.3 the score for a deletion would be set to be the same in column 2, where there is a deletion in one sequence, HBB\_HUMAN, as in column 4, where there is a deletion opening in five of the seven sequences. It would be more reasonable to set the probability of a new gap opening to be higher in column 4.

Changes have been made to non-probabilistic profiles to address these and

other  
1996

Let u  
emis  
zero,  
alpha  
seque  
peak

Th  
the v  
say al  
al. [1  
we w  
nique

Th  
which  
to ins

had a  
clear  
and ti  
insert  
rule t  
insert  
chara

Th  
align  
x to c  
direct  
times

MM

□

where  
probe  
Th In  
ment,  
ever,

that's  
and s



other problems [Thompson, Higgins & Gibson 1994b; Gribskov & Veretnik 1996], and we shall return to some of these later.

### Basic profile HMM parameterisation

Let us turn back to hidden Markov model profiles. Like all HMMs, these have emission and transition probabilities. Assuming that these probabilities are non-zero, a profile HMM can model any possible sequence of residues from the given alphabet. It therefore defines a probability distribution over the whole space of sequences. The aim of the parameterisation process is to make this distribution peak around members of the family.

The parameters we have available to control the shape of the distribution are the values of the probabilities, and also the length of the model. There is a lot to say about setting these optimally. We give here the basic methods from Krogh *et al.* [1994]. After sections on database searching and variants for local alignment, we will return to an extended discussion of alternative parameter estimation techniques.

The choice of length of the model corresponds more precisely to a decision on which multiple alignment columns to assign to match states, and which to assign to insert states. The profile HMM we derived above from the single sequence *y* had a match state for each residue  $y_i$ . However, looking at Figure 5.3 it seems clear that the consensus sequence for this region should only have eight residues, and that the two non-starred residues in GLB1\_GLYDI should be treated as an insertion with respect to the consensus. For the time being we will use a heuristic rule to decide which columns should correspond to match states, and which to inserts. A simple rule that works well is that columns that are more than half gap characters should be modelled by inserts.

The second problem is how to assign the probability parameters. We regard the alignment as providing a set of independent samples of alignments of sequences  $x$  to our HMM. Since the alignments are given, we can estimate the parameters directly using equations (3.18) from Section 3.3. We just count up the number of times each transition or emission is used, and assign probabilities according to

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad \text{and} \quad e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}$$

where  $k$  and  $l$  are indices over states, and  $a_{kl}$  and  $e_k$  are the transition and emission probabilities, and  $A_{kl}$  and  $E_k$  are the corresponding frequencies.

In the limit of having a very large number of sequences in our training alignment, this will give an accurate and consistent estimate of the probabilities. However, it has problems when there are only a few sequences. A major difficulty is that some transitions or emissions may not be seen in the training alignment, and so would acquire zero probability, which would mean they would never be

protein  
will be

tate and gap  
They set the  
substitution  
ent column.  
our example

penalties for  
gap (either  
erved in the

ation, and it  
of families,  
rongly con-  
information  
atrix as that  
cysteine (C)  
lumn for an  
om a single  
elihood of a

res for gaps  
gure 5.3 the  
re there is a  
is a deletion  
le to set the

s these and

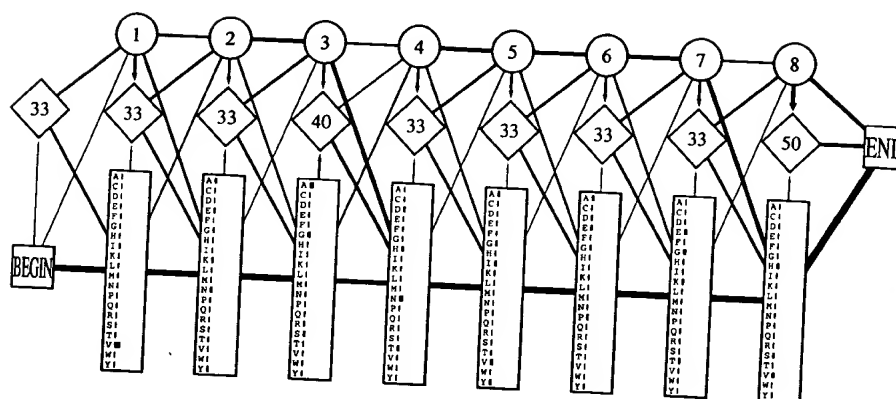


Figure 5.4 A hidden Markov model derived from the small alignment shown in Figure 5.3 using Laplace's rule. Emission probabilities are shown as bars opposite the different amino acids for each match state, and transition probabilities are indicated by the thickness of the lines. The  $I \rightarrow I$  transition probabilities times 100 are shown in the insert states. (Figure generated automatically using the SAM package.)

allowed in the future. More broadly, we are not using any previous knowledge about protein alignments, as the earlier non-probabilistic methods did implicitly, by using an independently derived substitution matrix. As a minimal approach to avoid zero probabilities, we can add pseudocounts to the observed frequencies (as in Chapters 1 and 3). The simplest pseudocount method is Laplace's rule: to add one to each frequency. We discuss better ways to choose the pseudocount values, and other approaches to estimating the parameters, at greater length below in Section 5.6.

#### Example: Parameters for an HMM based on Figure 5.3

Let us assume that we use Laplace's rule to obtain parameters for an HMM corresponding to the alignment in Figure 5.3. Then  $e_{M_1}(V) = 6/27$ ,  $e_{M_1}(I) = e_{M_1}(F) = 2/27$ , and  $e_{M_1}(a) = 1/27$  for all residue types  $a$  other than  $V$ ,  $I$ ,  $F$ . Similarly,  $a_{M_1M_2} = 7/10$ ,  $a_{M_1D_2} = 2/10$  and  $a_{M_1I_1} = 1/10$  (following column 1 there are six transitions from match to match, one transition to a delete state, in HBB\_HUMAN, and no insertions). Figure 5.4 shows the complete set of parameters for the HMM in diagrammatic form. □

#### 5.4 Searching with profile HMMs

One of the main purposes of developing profile HMMs is to use them to detect potential membership in a family by obtaining significant matches of a sequence to the profile HMM. We will assume for now that we are looking for global matches.

In prac  
more s  
We l  
can eit  
sequen  
calcula  
In ei  
uating  
probabi

We then  
designe  
log-odd  
could h  
cleaner  
units is  
we disc

Let  $V_j^M$   
the subn  
 $V_j^I(i)$  is  
the best

These ar  
 $e_{ij}(x_i)$  in  
from the  
bilities c  
may not i

In practice, as for pairwise alignment, one of the local alignment methods may be more sensitive for finding distant matches. We discuss these in the next section.

We have a choice of ways to score a match to a hidden Markov model. We can either use the Viterbi equations to give the most probable alignment  $\pi^*$  of a sequence  $x$  together with its probability  $P(x, \pi^* | M)$ , or the forward equations to calculate the full probability of  $x$  summed over all possible paths  $P(x | M)$ .

In either case, for practical purposes the result we want to consider when evaluating potential matches is the log-odds ratio of the resulting probability to the probability of  $x$  given our standard random model

$$P(x | R) = \prod_i q_{x_i}.$$

We therefore show here versions of the Viterbi and forward algorithms that are designed specifically for profile HMMs, and which result directly in the desired log-odds values. Note that changing to log-odds does not change the result; we could have subtracted the random model log score afterwards. However, it is cleaner and more efficient. Another practical reason for working in log-odds units is to avoid problems of underflow when working with raw probabilities, as we discussed in Section 3.6.

### Viterbi equations

Let  $V_j^M(i)$  be the log-odds score of the best path matching subsequence  $x_{1..i}$  to the submodel up to state  $j$ , ending with  $x_i$  being emitted by state  $M_j$ . Similarly  $V_j^I(i)$  is the score of the best path ending in  $x_i$  being emitted by  $I_j$ , and  $V_j^D(i)$  for the best path ending in state  $D_j$ . Then we can write

$$\begin{aligned} V_j^M(i) &= \log \frac{e_{M_j}(x_i)}{q_{x_i}} + \max \begin{cases} V_{j-1}^M(i-1) + \log a_{M_{j-1}M_j}, \\ V_{j-1}^I(i-1) + \log a_{I_{j-1}M_j}, \\ V_{j-1}^D(i-1) + \log a_{D_{j-1}M_j}; \end{cases} \\ V_j^I(i) &= \log \frac{e_{I_j}(x_i)}{q_{x_i}} + \max \begin{cases} V_j^M(i-1) + \log a_{M_jI_j}, \\ V_j^I(i-1) + \log a_{I_jI_j}, \\ V_j^D(i-1) + \log a_{D_jI_j}; \end{cases} \quad (5.1) \\ V_j^D(i) &= \max \begin{cases} V_{j-1}^M(i) + \log a_{M_{j-1}D_j}, \\ V_{j-1}^I(i) + \log a_{I_{j-1}D_j}, \\ V_{j-1}^D(i) + \log a_{D_{j-1}D_j}. \end{cases} \end{aligned}$$

These are the general equations. In a typical case, there is no emission score  $e_{I_j}(x_i)$  in the equation for  $V_j^I(i)$  because we assume that the emission distribution from the insert states  $I_j$  is the same as the background distribution, so the probabilities cancel in the log-odds form. Also, the  $D \rightarrow I$  and  $I \rightarrow D$  transition terms may not be present, as discussed above.

We need to take a little care over initialisation and termination of the dynamic programming. We want to allow the alignment to start and end in a delete or insert state, in case the beginning or end of the sequence does not match the first or the last match state of the model. The simplest way to ensure this mechanistically is to rename the Begin state as  $M_0$  and set  $V_0^M(0) = 0$  (as we did in Chapter 3). We then allow transitions to  $I_0$  and  $D_1$ . Similarly, at the end we can collect together possible paths ending in insert and delete states by renaming the End state to  $M_{L+1}$  and using the top relation without the emission term to calculate  $V_{L+1}^M(n)$  as the final score.

If these recurrence equations are compared with those for standard gapped dynamic programming in (2.16), it can be seen that apart from renaming of variables this is the same algorithm, but with the substitution, gap-open and gap-extend scores all depending on position in the model,  $j$ .

### Forward algorithm

The recurrence equations for the forward algorithm are similar to the Viterbi equations, but with the  $\max()$  operation replaced by addition. We define variables  $F_j^M(i)$ ,  $F_j^I(i)$  and  $F_j^D(i)$  for the partial full log-odds ratios, corresponding to  $V_j^M(i)$ ,  $V_j^I(i)$  and  $V_j^D(i)$ . The recurrence equations are then:

$$\begin{aligned} F_j^M(i) &= \log \frac{e_{M_j}(x_i)}{q_{x_i}} + \log [a_{M_{j-1}M_j} \exp(F_{j-1}^M(i-1)) \\ &\quad + a_{I_{j-1}M_j} \exp(F_{j-1}^I(i-1)) + a_{D_{j-1}M_j} \exp(F_{j-1}^D(i-1))]; \\ F_j^I(i) &= \log \frac{e_{I_j}(x_i)}{q_{x_i}} + \log [a_{M_{j-1}I_j} \exp(F_{j-1}^M(i-1)) \\ &\quad + a_{I_{j-1}I_j} \exp(F_{j-1}^I(i-1)) + a_{D_{j-1}I_j} \exp(F_{j-1}^D(i-1))]; \\ F_j^D(i) &= \log [a_{M_{j-1}D_j} \exp(F_{j-1}^M(i)) + a_{I_{j-1}D_j} \exp(F_{j-1}^I(i)) \\ &\quad + a_{D_{j-1}D_j} \exp(F_{j-1}^D(i))]. \end{aligned}$$

Initialisation and termination conditions are handled as for the Viterbi case, with  $F_0^M(0)$  being initialised to 0.

Although these appear a little complicated, in a practical implementation the operation  $\log(e^x + e^y)$  can be performed efficiently to adequate accuracy by function lookup and interpolation; see Section 3.6.

### Alternatives to log-odds scoring

In some of the earlier papers on HMMs, rather than calculating the log-odds score relative to a random model, the logarithm of the probability of the sequence given the model was used directly. This was called the LL score for 'log likelihood'.  $LL(x) = \log P(x|M)$ . The LL score is strongly length dependent, so for searching

LL/length

0  
-1  
-2  
-3  
-4  
-5  
-6  
F  
oj

it is no  
the seq  
betwee  
A w  
viation  
each se  
illustra

Examp  
From 3  
scratch  
in Chap  
done se  
(We use  
[1996])  
With  
Bairoch  
and log  
the ami  
the leng  
the othe  
to 300  
globins  
The  
globins  
clearer.  
A few

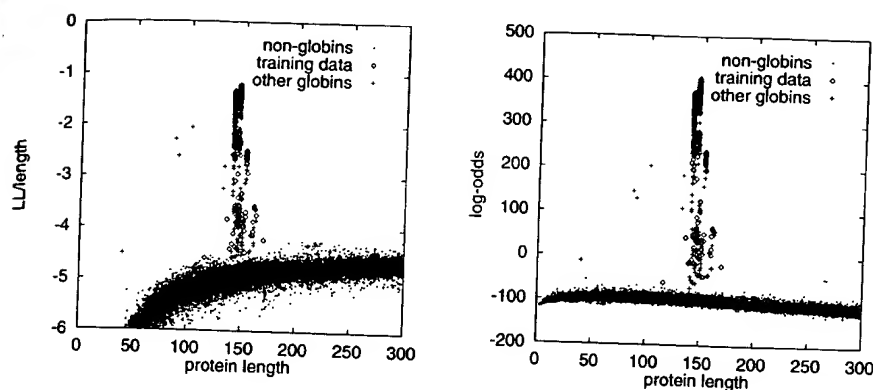


Figure 5.5 To the left the length-normalized LL score is shown as a function of sequence length. The right plot shows the same for the log-odds score.

it is not good enough to use a simple threshold. It is better to use LL divided by the sequence length, but even that is not always perfect, because the dependence between LL and sequence length is not linear (see example below).

A way to get around this is to estimate an average score and a standard deviation as a function of length and then use the number of standard deviations each sequence is away from the average. This is called the Z-score, and is also illustrated in the example below.

#### Example: Modelling and searching for globins

From 300 randomly picked globin sequences a profile HMM was estimated from scratch, i.e. starting from unaligned sequences using procedures we will explain in Chapter 6. A simple pseudocount regulariser was used. The estimation was done several times and the model with the highest overall LL score was picked. (We used the default settings of the SAM package, version 1.2; Hughey & Krogh [1996]).

With this model a database of about 60 000 proteins (SWISS-PROT release 34; Bairoch & Apweiler [1997]) was searched using the forward algorithm. The LL and log-odds scores were found for each sequence. For the null model we used the amino acid frequencies of the 300 sequences in the training set. In Figure 5.5 the length-normalised scores are shown for all the globins in the training set, all the other globins in the database and all the rest of the proteins with lengths up to 300 amino acids.<sup>1</sup> The globin sequences are clearly separated from the non-globins apart from a few in the 'twilight zone.'

The main difference between the two is in the variance of the score for non-globins, which is lower for the log-odds score, and therefore the separation is clearer. However, just choosing a cut-off of zero for the log-odds would miss a

A few dubious globins and other strange sequences were removed from these data.

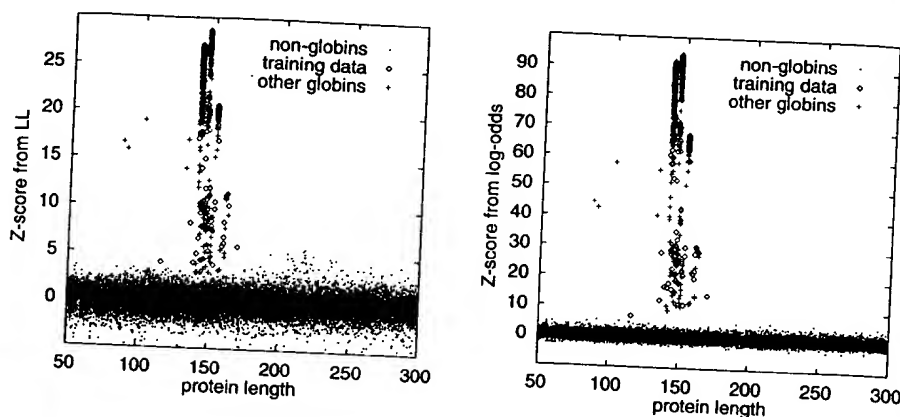


Figure 5.6 The Z-score calculated from the LL scores (left) and the log-odds (right).

lot of real globins in the search. This is because the profile HMM is not broad enough: it is too concentrated on a subset of the globins. Although there are ways to address this problem directly that we will return to later in the chapter, it is also possible to take a pragmatic approach to the separation of signal from noise given the results of the search, and calculate Z-scores for each hit.

To calculate Z-scores, a smooth curve is fitted to the LL or log-odds score of the non-globin sequences (a method is outlined in Krogh *et al.* [1994]). A standard deviation is then estimated for each length (or rather a little interval around it), and for each score the distance from the smooth curve is calculated in units of the standard deviation. This is the Z-score. The result (still as a function of sequence length) is shown in Figure 5.6.<sup>2</sup>

It is evident that it is now possible to find a threshold which will separate most globins from all other sequences. It is also clear that the score based on log-odds is much better for discrimination, with approximately three times the signal to noise ratio of the LL score. The reason for this is that dividing by the probability of the random model adjusts for the residue composition of the sequence. Without doing that, sequences with similar residue compositions as globins will tend to score more highly than sequences containing different residues, increasing the variance of the noise. □

## Alignment

Aside from finding matches, the other principal use of profile HMMs is to give an alignment of a sequence to the family; or more precisely to add it into the multiple alignment of the family. This is primarily the subject of the next chapter,

<sup>2</sup> There is no analytical result about the shape of these score distributions. The global alignment distribution is probably not exactly a Gaussian [Waterman 1995], but it appears to be a good approximation. For local alignments the extreme value distribution may be more reasonable, as discussed in Chapter 2.

on  
len  
est  
vari  
of (the

We l  
of a  
ison  
Chap  
gram  
HMM  
He  
Chap  
conve  
we as  
 $\sum_x P$   
compl  
one or  
region  
added  
used to  
The

The flar  
as speci  
of cours  
looping



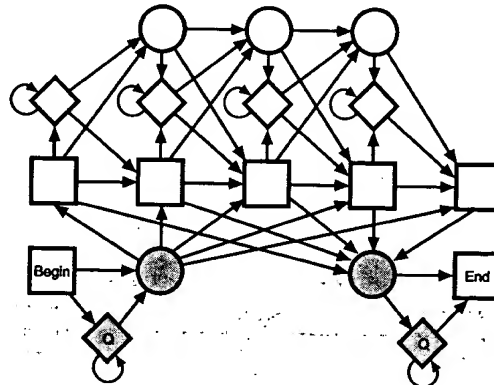
on multiple alignment methods, which covers alignment with profile HMMs at length. For now, we will just point out that the natural solution is to take the highest scoring, or Viterbi, alignment. This is obtained by tracing back on the Viterbi variables  $V_j^*(i)$ , exactly as with pairwise alignment. Beyond this, all the methods of Chapter 4 can be applied, to explore variants, and to assess the reliability of the alignment.

## 5.5 Profile HMM variants for non-global alignments

We have seen that there is a very close relationship between the Viterbi alignment of a sequence to a profile HMM and the global dynamic programming comparison between two sequences using affine gap penalties, which we described in Chapter 2. It is therefore possible to generalise all the variations of dynamic programming, such as those that find local, repeat and overlap matches, to use profile HMMs.

However, we have developed probabilistic models much more fully since Chapter 2, and this time we want to take more care to ensure that the result of converting to a local algorithm remains a proper probabilistic model, i.e. that we assign each sequence a true probability so that the sum over all sequences  $\sum_x P(x|M) = 1$ . Our approach to doing this is to specify a new model for the complete sequence  $x$ , which incorporates the original profile HMM together with one or more copies of a simple self-looping model that is used to account for the regions of unaligned sequence. These behave very like the insert states that we added to the profile itself. We call them *flanking model* states, because they are used to model the flanking sequences to the actual profile match itself.

The model for local (Smith–Waterman style) alignment is shown here:

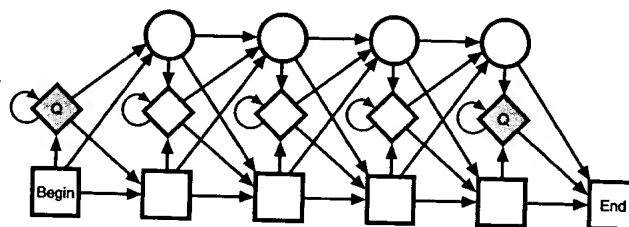


The flanking model states are shown as shaded diamonds. Notice that as well as specifying the emission probabilities of the new states, which will normally of course be  $q_a$ , we must specify a number of new transition probabilities. The looping probability on the flanking states should be close to 1, since they must

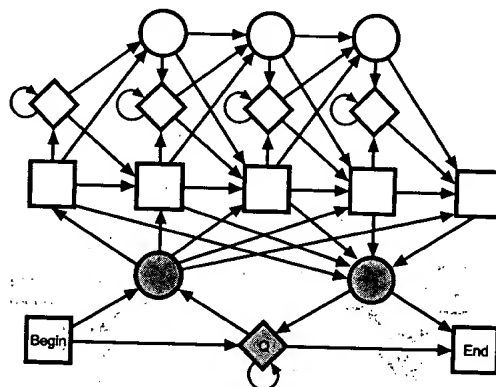
account for long stretches of sequence. Let us set these to  $(1 - \eta)$ . Note also that we have made use of silent states, shown as shaded circles, as 'switching points' to reduce the total number of transitions.

The next issue is how to set all the transition probabilities from the left flanking state to different start points in the model. One option is to give them equal probabilities,  $\eta/L$ . Another is to assign more probability to starting at the beginning of the model. The default option in the HMMER package for profile HMMs [Eddy 1996] assigns probability  $\eta/2$  to the start of the profile, and  $\eta/(2(L - 1))$  to the other positions, favouring matches that start at the beginning of the model.

If all the probability is assigned to the first model state, then it forces this model to match only complete copies of the profile in the searched sequence, ensuring a type of 'overlap' match constraint. This can be appropriate when, for example, the HMM represents a protein domain that you expect to find either present as a whole or absent. However, to allow for rare cases where the first residue might be missing, it may be wise in such cases to allow a direct transition from the flanking state into a delete state, as shown here:



It is clear that by tinkering with the transition connections and probabilities a wide variety of different models can be produced, each potentially useful in different circumstances. A final example similar to the first model for local matches is



which allows repeat matches to subsections of the profile model, like the repeat algorithm variant in Chapter 2.

Note that all these variants of transition connectivity and probability assignment affect not only the types of match that are allowed, but also the score. More

restrict  
found  
match

Exer  
5.1

5.2

As pr  
great  
on the  
proba  
tailed  
count  
The  
the m  
slight  
positi  
spond

As  
exam  
will f  
F are  
will c  
easier  
count  
then g

A ver  
counts  
one, :



restrictive transition distributions will give higher match scores if a good match is found, so are preferable if they can be designed to represent the types of correct matches that are expected.

### Exercises

- 5.1 Show that if the random model is the same as that described in Chapter 4 (a succession of two states looping on themselves with probability  $(1 - \eta)$ ), with  $\eta$  the same as in the flanking models, the local alignment model gives update equations like those of equation (2.9).
- 5.2 Explain the reasons for any differences.

## 5.6 More on estimation of probabilities

As promised above, we now return to the subject of parameter estimation at greater length. Although our discussion for most of this section will be focused on the emission probabilities, analogous methods can be used for the transition probabilities. The aim here is to introduce methods that can be used. A more detailed mathematical discussion about the estimation of probabilities from sample counts is given in Chapter 11 (p. 311).

The most straightforward approach to parameter estimation would be to give the maximum likelihood estimates for the parameters. We will change notation slightly from that used before. Given observed frequencies  $c_{ja}$  of residue  $a$  in position  $j$  of the alignment, maximum likelihood estimates for  $e_{M_j}(a)$ , the corresponding model parameters, are

$$e_{M_j}(a) = \frac{c_{ja}}{\sum_{a'} c_{ja'}}. \quad (5.2)$$

As we described above, a clear problem with this is that if there are no observed examples of a particular outcome then its probability is estimated as zero. This will frequently occur. For example, in the alignment of Figure 5.3 only V, I and F are present in the first column. However, it is quite likely that other amino acids will occur in that position amongst all the other globin sequences in biology. The easiest way to deal with this problem is to add pseudocounts to the observed counts  $c_{ja}$ . Below, we first discuss the pseudocount approach at greater length, then give some more complex alternatives.

### Simple pseudocounts

A very simple and much-used pseudocount method is to add a constant to all the counts, which prevents the problem with zero probabilities. When the constant is one, as we used above in our example, this is called 'Laplace's rule'. A slightly

more sophisticated method is to add a quantity proportional to the background distribution, giving

$$e_{M_j}(a) = \frac{c_{ja} + Aq_a}{\sum_{a'} c_{ja'} + A}, \quad (5.3)$$

where  $c_{ja}$  are the real counts, and  $A$  is the weight put on the pseudocounts as compared to the real counts. Values of  $A$  of around twenty seem to work well for protein alignments.

This form of regularisation has the appealing feature that  $e_{M_j}(a)$  is approximately equal to  $q_a$  if very little data is available, i.e. all the real counts are very small compared to  $A$ . At the other extreme, where a large amount of data is available, the effect of the regulariser becomes insignificant and  $e_{M_j}(a)$  is essentially equal to the maximum likelihood solution. So, at this intuitive level, pseudocounts make a lot of sense.

Adding pseudocounts amounts to adding some fake imagined data into the alignment, based on our general knowledge of proteins, to represent all the other things that might happen. They thus correspond to prior information about protein families, before having seen the specific data for the family in the form of the alignment. This statement can be formalised in a Bayesian framework. Bayes' equation tells us how to combine data,  $D$ , with a prior probability distribution over the parameters  $P(\theta)$  to give a posterior distribution over  $\theta$ , from which we can take either the maximum or the mean as our best estimate,

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}.$$

In our case the parameters  $\theta$  are our model probabilities. The pseudocount method given above corresponds in this Bayesian framework to assuming a Dirichlet prior distribution with parameters  $\alpha_a = Aq_a$  over the probabilities; see Chapter 11 for mathematical details.

### Dirichlet mixtures

The problem with the simple pseudocounts, as compared to the substitution matrix based methods, is that only the most rudimentary prior knowledge can be contained in a single pseudocount vector. For this reason we need a lot of example data in the alignment to get good estimates of the parameters. Experience suggests that to achieve good discrimination typically fifty or more examples are desirable when modelling proteins.

In order to include better prior information, it was therefore suggested by Brown *et al.* [1993] that one should use a *mixture* of Dirichlet distributions as the prior. The idea is that there might be several different sets of pseudocount priors  $\alpha^1, \dots, \alpha^K$  corresponding to different types of alignment environments, where

$\alpha_a^k$  corresponds to  $Aq_a$  in the example above. One set might be relevant for exposed loop environments, one for buried small residue environments, etc. Given our counts  $c_{ja}$  we first estimate how likely each prior distribution  $k$  is (based on how well it fits the observed data), then combine their effects according to these posterior probabilities:

$$e_{M_j}(a) = \sum_k P(k|c_j) \frac{c_{ja} + \alpha_a^k}{\sum_{a'} (c_{ja'} + \alpha_{a'}^k)},$$

where the  $P(k|c_j)$  are the *posterior mixture coefficients*. We calculate these by Bayes' rule,

$$P(k|c_j) = \frac{p_k P(c_j|k)}{\sum_{k'} p_{k'} P(c_j|k')}$$

where the  $p_k$  are the prior probabilities of each mixture component, and  $P(c_j|k)$  is the probability of the data according to Dirichlet mixture  $k$ . The equation for  $P(c_j|k)$  has a frightening looking form, which is in fact fairly simple to calculate:

$$P(c_j|k) = \frac{(\sum_a c_{ja})!}{\prod_a c_{ja}!} \frac{\Gamma(\sum_a c_{ja} + \alpha_a^k)}{\prod_a \Gamma(c_{ja} + \alpha_a^k)} \frac{\Gamma(\sum_a \alpha_a^k)}{\prod_a \Gamma(\alpha_a^k)},$$

where  $\Gamma(x)$  is the gamma function, a standard function over the reals related to the factorial function on the integers. For further details and an explanation of this equation, see Chapter 11, where we also describe how the mixture component distributions  $\alpha_a^k$  are obtained.

Using this type of approach, it seems that good profile HMMs can be fit to alignments with as few as ten or twenty examples [Sjölander *et al.* 1996].

### Substitution matrix mixtures

An alternative approach to using a mixture of Dirichlets is to adjust the pseudocounts in a single Dirichlet formulation, using information from the observed counts and a substitution matrix. This is not a theoretically well-founded approach, but it makes intuitive sense as a heuristic, combining features of the non-probabilistic profile methods and the Dirichlet pseudocount methods.

The first step is to convert the matrix entries  $s(a,b)$  into conditional probabilities  $P(b|a)$ . If we assume that the substitution matrix entries are derived as log-odds ratios, as in Chapter 2, then  $s(a,b) = \log(P(a,b)/q_a q_b)$ , which is the same as  $\log(P(b|a)/P(b))$ , so  $P(b|a) = q_b e^{s(a,b)}$ . We can in fact derive  $P(b|a)$  values from an arbitrary score matrix  $s(a,b)$  given background probabilities  $q_a$ ; see below.

Given conditional probabilities  $P(b|a)$  we can generate pseudocounts as follows. Let  $f_{ja}$  be the maximum likelihood probabilities derived from the counts,

so  $f_{ja} = c_{ja} / \sum_a' c_{ja'}$ . Using these we set pseudocount values with

$$\alpha_{ja} = A \sum_b f_{jb} P(a|b),$$

where  $A$  is a positive constant comparable to the one we used with simple pseudocounts [Tatusov, Altschul & Koonin 1994; Claverie 1994; Henikoff & Henikoff 1996]. We then use essentially the same equation as (5.3) to obtain the model parameters:

$$e_{M_j}(a) = \frac{c_{ja} + \alpha_{ja}}{\sum_{a'} c_{ja'} + \alpha_{ja'}}.$$

There is no obvious statistical interpretation for this type of pseudocount, but the idea is quite natural: amino acid  $i$  contributes to pseudocount  $j$  in proportion to its abundance in the column and the probability of its changing to amino acid  $j$ . The formula interpolates between the treatment of pairwise alignments and the maximum likelihood solution. The substitution matrix term dominates if there are small numbers of sequences (especially if  $A \gg 1$ ), and values close to the maximum likelihood estimate are obtained when the number of counts is large (more precisely when the total number of counts  $C_j \gg A$ ).

There are various choices for the scaling constant  $A$  of the pseudocounts. For instance  $A = 1$  was used in Lawrence *et al.* [1993], but this appears to be too weak in practice. Claverie [1994] suggests  $A = \min(20, N)$ , and Henikoff & Henikoff [1996] suggest  $A = 5R$ , where  $R$  is the number of different residue types observed in the column (i.e. the number of  $a$  for which  $c_{ja} > 0$ ).

#### *Deriving $P(b|a)$ from an arbitrary matrix*

Even if a score matrix  $s(a, b)$  was not derived as a log-odds matrix, as long as certain conditions are fulfilled it is possible to find a scale factor  $\lambda$  such that  $\lambda s(a, b)$  will behave correctly when interpreted as a log-odds matrix [Altschul 1991]. The conditions are that the matrix is negatively biased, i.e.  $\sum_{ab} q_a q_b s(a, b) < 0$ , and that it contains at least one positive entry.

What we want is a set of values  $r_{ij}$  for which

$$s(a, b) = \frac{1}{\lambda} \log \frac{r_{ab}}{q_a q_b},$$

where  $r_{ab}$  can be interpreted as the probability for the pair  $a, b$ . This equation is easily inverted, so we get the pair probabilities expressed in terms of the substitution matrix  $r_{ab} = q_a q_b \exp(\lambda s(a, b))$ . To be legitimate probabilities the  $r_{ab}$  have to sum to one. We therefore need to find a  $\lambda$  such that

$$f(\lambda) = \sum_{a,b} q_a q_b e^{\lambda s(a,b)} = 1. \quad (5.4)$$

One such value is  $\lambda = 0$ , but clearly this is not what we want. The two conditions

we gave above turn out to be sufficient to ensure there is another, positive solution to this equation; see the exercises below.

The resulting value of  $\lambda$  is called the natural scaling factor of the substitution matrix. This probabilistic interpretation of the substitution matrix leads to an entropy measure for the matrix of  $\sum_{ab} r_{ab} \log(r_{ab}/q_a q_b)$ , which is a useful quantity for characterising and comparing substitution matrices [Altschul 1991].

### Exercises

- 5.3 Use the negative bias condition to show that  $f(\lambda)$  is negative for small enough  $\lambda$ . Hint: calculate  $f'(0)$ , the derivative of  $f(\lambda)$  at  $\lambda = 0$ .
- 5.4 Use the second condition, that there is at least one positive  $s(a, b)$ , to show that  $f(\lambda)$  becomes positive for large enough  $\lambda$ .
- 5.5 Finally, show that the second derivative of  $f(\lambda)$  is positive, and from this and the results of the previous two exercises that there is one and only one positive value of  $\lambda$  satisfying (5.4).

### Estimation based on an ancestor

There is a more principled and direct way to use the information in substitution matrices for estimating the HMM probabilities than that described above. This approach does not use pseudocounts. Instead, it assumes that all the observed sequences have been derived independently from a common ancestor, and generates an estimate of the residue present in a given position in that common ancestor (or rather a posterior probability distribution for what that residue was). From this we can estimate the probability of seeing each residue in a new descendant of the ancestor, different from those in the sample.

Assume we have example sequences  $x^k$  with residues  $x_j^k$  in column  $j$  of the alignment (we have adjusted our notation slightly; this  $x_j^k$  is not the  $j$ th residue in sequence  $x^k$  if there are gaps, but it is a convenient notation for what we need here). Once again, we need the conditional probabilities  $P(b|a)$  derived from the substitution matrix. Let the residue in the common ancestor be  $y_j$ . Then we can use Bayes rule to calculate the posterior probability that  $y_j = a$

$$P(y_j = a | \text{alignment}) = \frac{q_a \prod_k P(x_j^k | a)}{\sum_{a'} q_{a'} \prod_k P(x_j^k | a')}. \quad (5.5)$$

Note that we needed a prior distribution for residues at the common ancestor, which we set to  $q_a$  because that is our background probability for amino acids in the absence of further information.

We can now calculate the HMM emission probabilities as the predicted probabilities for a new sequence

$$e_{M_j}(a) = \sum_{a'} P(a|a') P(\bar{y}_j = a' | \text{alignment}). \quad (5.6)$$

One problem with this approach is that, as we noticed above, different columns vary widely in their degree of conservation. Indeed, that is one of the properties that we wanted to exploit when using alignments to estimate profile HMMs. However, using a single substitution matrix implies assuming a fixed degree of conservation. As we discussed in Chapter 2, matrices typically come in families varying in their level of implied conservation. Examples are the PAM [Dayhoff, Schwartz & Orcutt 1978] and the BLOSUM [Henikoff & Henikoff 1992] series of matrices. We can therefore significantly improve the approach in (5.5) and (5.6) if we optimise over choice of matrix from a family. This way, a very conserved column might use a conservative matrix, such as PAM30, and a very varied column would use a divergent matrix, such as PAM500.

How do we choose the optimal matrix? A natural approach is to maximise the likelihood of the observed data

$$P(x_j^1, \dots, x_j^N | t) = \sum_a q_a \prod_k P(x_j^k | a, t) \quad (5.7)$$

where  $t$  is the matrix family parameter ( $t$  for evolutionary *time*). It would also be possible to use a Bayesian approach here, proposing a prior distribution over  $t$ , then combining this with (5.7) in Bayes' rule to obtain a posterior distribution for  $t$ , and summing over this in (5.6). However, that would require significantly more computation.

The maximum likelihood time-dependent approach is closely related to the 'evolutionary weights' method in the PROFILE package [Gribskov & Veretnik 1996]. However, that method estimates different evolutionary times  $t$  for each possible ancestral amino acid, and also adjusts the resulting weights with respect to a set of baseline probabilities; for details see Gribskov & Veretnik [1996]. There are also strong connections between the methods of this subsection and those discussed later in Chapter 8 when building phylogenetic trees using maximum likelihood methods.

### Testing the pseudocount methods

All the methods mentioned above have been tested in various ways. Direct tests, in which profiles were constructed and used for searching, were carried out extensively by Henikoff & Henikoff [1996]. The best method turned out to be the substitution matrix based method (5.6), with  $A = 5R$  as described above; the Dirichlet mixture regulariser came a reasonably close second. Other tests gave different results [Tatusov, Altschul & Koonin 1994; Karplus 1995], so it is not clear which method is best, and it is likely that this will depend on the application and the details of the mixture components or substitution matrix used.

An interesting method for testing various regularisers was given by Karplus [1995]. Instead of performing a huge number of database searches, he

ask  
pro  
dee  
we  
cou  
othe  
des  
is d  
the  
do  
proc  
like

Kar  
mor  
whic  
liho  
to de

K  
resul  
ence  
we a  
obtai  
the t  
ones

Th  
sizes  
it is  
sizes  
the a

As  
free  
num  
the a  
some  
the r  
mini  
et al.  
be de



asked the following question: How well can an amino acid distribution be approximated from a small sample? Columns were extracted from a large set of deep alignments (the BLOCKS database; Henikoff & Henikoff [1991]). Imagine we take a small sample of size  $n$  with counts  $c_a$  from a column with complete counts  $C_a$ . From the sample counts  $c_a$  we can estimate the frequencies  $e_c(a)$  of other symbols that might occur in the same column, using one of the methods described above (we use a subscript  $c$  to remind ourselves that this estimation is dependent on the sample counts). We can now calculate the log likelihood of the other symbols that actually do occur, as  $\sum_a (C_a - c_a) \log e_c(a)$ . Note that we do not score the counts that were used in the sample. We can now repeat this process for all samples of size  $n$  from all columns and sum all the resulting log likelihoods,

$$(5.7) \quad LL = \sum_{\text{columns } C} \sum_{\text{samples } c} \sum_a (C_a - c_a) \log e_c(a). \quad (5.8)$$

Karplus proposed that a good regulariser should maximise this value. Furthermore, he pointed out that there is clearly an optimal strategy for such a process, which is to tabulate for each possible set of sample counts  $c_a$  the maximum likelihood distribution emission distribution  $e_c(a)$ . This is only practically possible to do explicitly up to  $n = 5$ .

Karplus showed that several of the more complex regularisers described above resulted in estimators that were very close to optimal, in the sense that the difference in LL values from optimal was very small at  $n = 5$ . Of course, ultimately we are interested in database searches, and it is not evident that the regulariser obtaining the lowest LL score will actually be best for searching. It is likely that the typical similarities in the source alignment database are not the same as the ones that we will be searching for with our HMM.

The actual averaging over all samples can only be done explicitly for sample sizes up to around  $n = 5$ , but that is also the most interesting regime, because it is for small sample sizes that regularisation is most crucial. For larger sample sizes we would have to use some form of random sampling method to estimate the average.

As well as evaluating methods, Karplus' approach can also be used to set the free parameters in the various methods described above, for example the total number of pseudocounts  $A$  to use in (5.3). For any value of  $A$  we can calculate the average log likelihood from our database of columns, either directly or by some sort of random sampling, and in fact we can also calculate the gradient of the relative entropy with respect to  $A$ . We can therefore find the value of  $A$  that minimises this average relative entropy, using gradient descent methods [Press *et al.* 1992], or by other optimisation methods. Of course, in principle this can be done for any sample size  $n$ , yielding parameters dependent on  $n$ . However,

because the averaging is difficult for  $n > 5$ , this technique has yet to prove its potential.

## 5.7 Optimal model construction

When we first discussed the parameterisation of profile HMMs, we pointed out that as well as estimating the probability parameters, it is necessary to decide which columns of the alignment should be assigned to insert states, and which to match states. We call this process model construction. At the time we proposed a simple heuristic, but we can do better than that. There is an efficient dynamic programming algorithm which can find the column assignments that maximise the posterior probability of the mode, at the same time as fitting optimal probability parameters.

In the profile HMM formalism, it is assumed that an aligned column of symbols corresponds either to emissions from the same match state or to emissions from the same insert state. It therefore suffices to mark which columns come from match states to specify a profile HMM architecture and the state paths for all the sequences in the alignment, as shown in Figure 5.7. In a marked column, symbols are assigned to match states and gaps are assigned to delete states. In an unmarked column, symbols are assigned to insert states and gaps are ignored. State transition and symbol emission counts are obtained from the state paths, and these counts can be used to estimate probability parameters by one of the methods in the previous section. In passing, we note that this model estimation procedure implicitly assumes that the multiple alignment is correct, i.e. that the implied state paths have probability one and all other state paths have probability zero, which is akin to a Viterbi assumption. The next chapter addresses issues of simultaneous alignment and model estimation.

There are  $2^L$  combinations of markings for an alignment of  $L$  columns, and hence  $2^L$  different profile HMMs to choose from. There are at least three ways to determine the marking. In *manual* construction, the user marks alignment columns by hand. This is perhaps the simplest way to allow users to manually specify the model architecture to use for a given alignment. In *heuristic* construction, a rule is used to decide whether a column should be marked. For instance, a column might be marked when the proportion of gap symbols in it is below a certain threshold. In *MAP* construction, a maximum *a posteriori* choice is determined by dynamic programming. A description of this algorithm follows.

### MAP match-insert assignment

The MAP construction algorithm recursively calculates a number  $S_j$ , which is the log probability of the optimal model for the alignment up to and including column

(b)



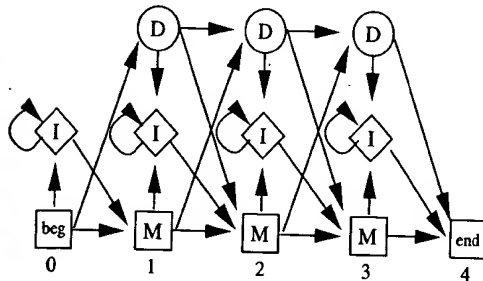
j, as  
endin  
proba  
releva  
are in  
colum  
Tra  
colum  
thus n  
are cc  
marke  
a sing  
For  
betwe  
transit



(a) Multiple alignment:

	x	x	.	.	x
bat	A	G	-	-	C
rat	A	-	A	G	-
cat	A	G	-	A	A
gnat	-	-	A	A	A
goat	A	G	-	-	C
	1	2	.	.	3

(b) Profile-HMM architecture:



(c) Observed emission/transition counts

		model position			
		0	1	2	3
match emissions	A	-	4	0	0
	C	-	0	0	4
	G	-	0	3	0
	T	-	0	0	0
insert emissions	A	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
state transitions	M-M	4	3	2	4
	M-D	1	1	0	0
	M-I	0	0	1	0
	I-M	0	0	2	0
	I-D	0	0	1	0
	I-I	0	0	4	0
	D-M	-	0	0	1
	D-D	-	1	0	0
	D-I	-	0	2	0

**Figure 5.7** As an example of model construction from an alignment, a small DNA multiple alignment is given (a), with three columns marked above with *x*'s. These three columns are assigned to positions 1–3 in the model architecture (b). The assignment of columns to model positions determines the symbol emission and state transition counts (c) from which probability parameters would be estimated.

$j$ , assuming that column  $j$  is marked.  $S_j$  is calculated from smaller subalignments ending at a marked column  $i$  ( $i < j$ ) by incrementing  $S_i$  with the summed log probability of the transitions and emissions for the columns between  $i$  and  $j$ . The relevant probability parameters are estimated 'on the fly' from the counts that are implied by marking columns  $i$  and  $j$  while leaving unmarked the intervening columns (if any).

Transition and emission counts for a section of alignment bounded by marked columns  $i$  and  $j$  are independent of how columns are marked before  $i$  and after  $j$ , thus making a dynamic programming recursion possible. Only marked columns are considered in the recursion, because transition and emission counts for unmarked columns are not independent of the assignment of neighbouring columns; a single insert state may account for more than one column in the alignment. For instance, let  $\mathcal{T}_{ij}$  be the summed log probability of all the state transitions between marked columns  $i$  and  $j$ . We can determine  $\mathcal{T}_{ij}$  from the observed state transition counts  $c_{xy}$  and the probabilities  $t_{xy}$ :

$$\mathcal{T}_{ij} = \sum_{x,y \in \{M,D,I\}} c_{xy} \log a_{xy}.$$

where  $a_{xy}$  is the probability of a state transition from  $x$  to  $y$ .

Transition counts  $c_{xy}$  are obtained from the partial state paths implied by marking  $i$  and  $j$ . For instance, if in one sequence we see a gap in column  $i$ , five residues in columns  $i + 1$  to  $j - 1$ , and a residue in column  $j$ , we would count one delete-insert transition, four insert-insert transitions, and one insert-match transition. The transition probabilities  $a_{xy}$  are estimated from the  $c_{xy}$  in the usual fashion, possibly including Dirichlet prior terms  $\alpha_{xy}$  (or indeed, any form of prior that is independent of the marking outside of  $i, \dots, j$ ):

$$a_{xy} = \frac{c_{xy} + \alpha_{xy}}{\sum_y c_{xy} + \alpha_{xy}}.$$

Let  $\mathcal{M}_j$  be the analogous log probability contribution for match state symbol emissions in column  $j$ , and  $\mathcal{I}_{i+1,j-1}$  be the same for the insert state emissions for columns  $i + 1, \dots, j - 1$  (for  $j - i > 1$ ). We can now give the algorithm:

**Algorithm: MAP model construction**

Initialisation:

$$S_0 = 0, \mathcal{M}_{L+1} = 0.$$

Recurrence: for  $j = 1, \dots, L + 1$ :

$$S_j = \max_{0 \leq i < j} S_i + \mathcal{I}_{ij} + \mathcal{M}_j + \mathcal{I}_{i+1,j-1} + \lambda;$$

$$\sigma_j = \operatorname{argmax}_{0 \leq i < j} S_i + \mathcal{I}_{ij} + \mathcal{M}_j + \mathcal{I}_{i+1,j-1} + \lambda.$$

Traceback: From  $j = \sigma_{L+1}$ , while  $j > 0$ :

Mark column  $j$  as a match column;

$$j = \sigma_j.$$

A profile HMM is then built from the marked alignment. The extra term  $\lambda$  is a penalty used to favour models with fewer match states. In Bayesian terms,  $\lambda$  is the log of the prior probability of marking each column, implying a simple but adequate exponentially decreasing prior distribution over model lengths.

With some care in implementation, this algorithm is  $O(L)$  in memory and  $O(L^2)$  in time for an alignment of  $L$  columns.

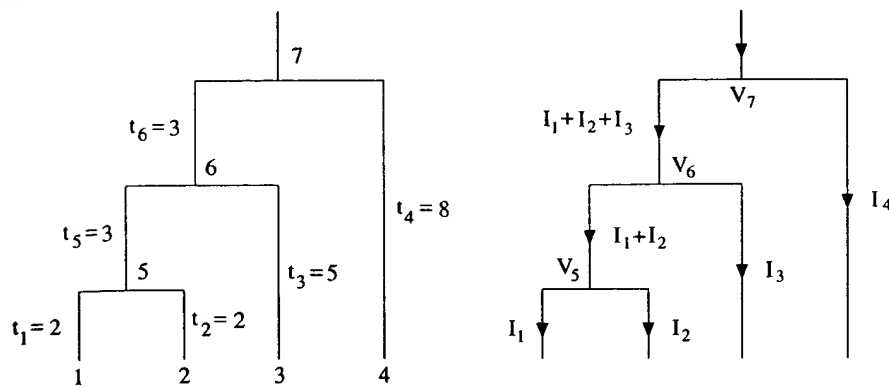
## 5.8 Weighting training sequences

One issue that we have avoided completely so far is that of weighting sequences when estimating parameters. In a typical alignment, there are often some sequences that are very closely related to each other. Intuitively, some of the information from these sequences is shared, so we should not give them each the same influence in the estimation process as a single sequence that is more highly diverged from all the others. In the extreme that two sequences are identical, it makes sense that they should each get half the weight of other sequences, so that

$t_1 = 2$

the n  
ically  
sequ  
To de  
diffe  
used  
mod

Man  
Since  
proac  
contr  
have  
later  
align  
not t  
sequ  
each  
for Or  
son 1  
cond  
leave  
and t  
vided



**Figure 5.8** On the left, a tree of sequences with branch lengths. On the right, the corresponding 'current' and 'voltage' values used in the 'Kirchhoff's law' approach to sequence weighting (see text).

the net effect is of having only one of them. Statistically, the problem is that typically the examples we have do not constitute a good random sample from all the sequences that belong to the family; the assumption of independence is incorrect. To deal with this sort of situation, there have been a large number of proposals for different ways to assign weights to sequences. In principle, any of these can be used in combination with any of the methods of the preceding sections on fitting model parameters and model construction.

### Simple weighting schemes derived from a tree

Many weighting approaches are based on building a tree relating the sequences. Since sequences in a family are related by an evolutionary tree, a very natural approach is to try to reconstruct this tree and use it when estimating the independent contribution of each of the observed sequences, downweighting sequences that have only recently diverged. We discuss phylogenetic tree construction at length later in Chapters 7 and 8, as well as in the next chapter on multiple sequence alignment. For our current purposes, the fine details of the method are probably not too important, and we will assume that we are given a tree connecting the sequences, with branch lengths indicating the relative degrees of divergence for each edge in the tree.

One of the intuitively simplest weighting schemes [Thompson, Higgins & Gibson 1994b] can be expressed nicely as follows. We are given a tree made of a conducting wire of constant thickness and apply a voltage  $V$  to the root. All the leaves are set to zero potential and the currents flowing from them are measured and taken to be the weights. Clearly, the currents will be smaller in the highly divided parts of the tree so these weights have the right qualitative properties. They

can be calculated by applying Kirchhoff's laws. For instance, in the tree shown in Figure 5.8, let the current and voltage at node  $n$  be  $I_n$  and  $V_n$ , respectively. Since constant factors do not affect the calculation, we can set the resistance equal to the edge-time. We then find  $V_5 = 2I_1 = 2I_2$ ,  $V_6 = 2I_1 + 3(I_1 + I_2) = 5I_3$ , and  $V_7 = 8I_4 = 5I_3 + 3(I_1 + I_2 + I_3)$ . There are therefore three equations relating the four currents, and these give  $I_1 : I_2 : I_3 : I_4 = 20 : 20 : 32 : 47$ .

Another attractively simple idea was proposed by Gerstein, Sonnhammer & Chothia [1994]. Their algorithm works up the tree from the leaves, incrementing the weights. Initially the weight of a sequence is set equal to the edge-time of the edge immediately above it. Now, suppose node  $n$  has been reached. The edge above  $n$  has edge-time  $t_n$ , and this is shared out amongst the weights of all the sequences at the leaves below  $n$ , incrementing them by a fraction proportional to their current weight values. Formally, the increase  $\Delta w_i$  in a weight  $w_i$  is given by

$$\Delta w_i = t_n \frac{w_i}{\sum_{\text{leaves } k \text{ below } n} w_k}. \quad (5.9)$$

The same operation is carried out up to the root.

This is clearly an easy and efficient algorithm. For instance, the weights in the tree of Figure 5.8 are computed as follows: Initially the weights are set to the edge lengths of the leafs,  $w_1 = w_2 = 2$ ,  $w_3 = 5$ , and  $w_4 = 8$ . At node 5 the edge length of 3 above node 5 is shared out equally to  $w_1$  and  $w_2$ , giving them  $3/2$  each, so now  $w_1 = w_2 = 2 + 3/2 = 3.5$ . At node 6 we find the edge of length 3 above node 6 is shared out to nodes 1, 2 and 3 in the ratio  $3.5 : 3.5 : 5$ , making  $w_1 = w_2 = 3.5 + 3 \times 3.5/12$ , and  $w_3 = 5 + 3 \times 5/12$ . With  $w_4 = 8$ , this gives  $w_1 : w_2 : w_3 : w_4 = 35 : 35 : 50 : 64$ . Even though these weights are close to those given by the Kirchhoff rule, the methods are in a sense opposed, for in a tree with two leaves and one edge longer than the other, the longer edge is down weighted by Kirchhoff and up weighted by (5.9).

### Root weights from Gaussian parameters

One view of weights is that they should represent the influence of leaves on the root distribution. It is possible to make this idea precise, as Altschul, Carroll & Lipman [1989] showed. They built on the version of Felsenstein's 'pruning' algorithm which applies to continuous parameters [Felsenstein 1973]. Instead of discrete members of an alphabet we have a continuous real-valued variable, like the weight of an organism. In place of a substitution matrix we have a probability density that defines the probability of substituting one value,  $x$ , of this variable by another,  $y$ . A simple example of such a density is a Gaussian, where the probability of  $x \rightarrow y$  along an edge with time  $t$  is  $\exp(-(x - y)^2 / (2\sigma^2 t))$ . The

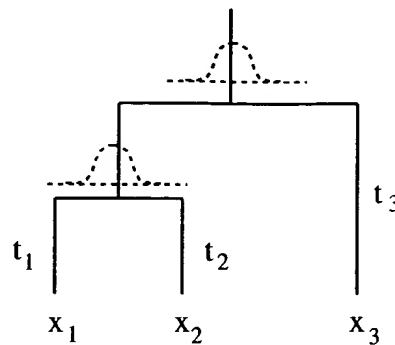


Figure 5.9 The tree described in the text when deriving Gaussian weights.

pruning algorithm now proceeds exactly as for a finite alphabet, but with integrals replacing discrete sums [Felsenstein 1973].<sup>3</sup>

Felsenstein's algorithm yields a Gaussian distribution for the parameter in question at the root whose mean  $\mu$  depends linearly on the values  $x_i$  of the parameters at the leaves, so  $\mu = \sum w_i x_i$ . Altschul, Carroll & Lipman [1989] proposed that these  $w_i$  should be used as weights. They represent the influence of each leaf at the root.

#### Example: Altschul–Carroll–Lipman weights for a three-leaf tree

To illustrate how the weights are derived, consider the simple three-leaf tree shown in Figure 5.9, where leaf  $i$  takes the value  $x_i$ . The probability distribution at node 4 is given by

$$P(x \text{ at node 4} \mid L_1, L_2) = K_1 e^{-\frac{(x-x_1)^2}{2t_1}} e^{-\frac{(x-x_2)^2}{2t_2}}$$

where  $K_1$  is a normalising constant. One can rewrite this as

$$P(x \text{ at node 4} \mid L_1, L_2) = K_1 e^{-\frac{(x-v_1x_1-v_2x_2)^2}{2t_{12}}}$$

where  $v_1 = t_2/(t_1+t_2)$ ,  $v_2 = t_1/(t_1+t_2)$  and  $t_{12} = t_1t_2/(t_1+t_2)$ . If we were considering only the two-leaf tree with root at node 4, the mean of the root distribution would be given by  $\mu = v_1x_1 + v_2x_2$ , and the weights would be  $v_1$  and  $v_2$ . Continuing with our three-leaf tree, however, we find next that the distribution at node 5

<sup>3</sup> Historically, the continuous case came first, and Felsenstein defined the pruning algorithm for Gaussian distributions of real-valued parameters. In the cited paper he takes account of the distribution of the parameters at each leaf, e.g. the mean and variance of the weight of an organism. Puzzlingly, he also introduces covariances between values for different leaves.

It is not clear how to calculate a covariance between, say, the weights of cows and cats. For proteins, having multiple corresponding sites in an alignment would allow correlations to be considered in principle.

is given by

$$P(y \text{ at node 5} \mid L_1, L_2, L_3) = K_2 e^{-\frac{(y-x_3)^2}{2t_3}} \int e^{-\frac{(x-v_1x_1-v_2x_2)^2}{2t_{12}}} e^{-\frac{(x-y)^2}{2t_4}} dx$$

where  $K_2$  is a normalising constant, and the integral is taken over all possible values of  $x$  at node 4 (and is the exact equivalent of the sum over all possible ancestral assignments of residues in the case of a discrete alphabet). This is a standard Gaussian integral, and boils down to the following

$$P(y \text{ at node 5} \mid L_1, L_2, L_3) = K_3 e^{-\frac{(y-w_1x_1-w_2x_2-w_3x_3)^2}{2t_{123}}}$$

where  $K_3$  is a new normalising constant and  $t_{123} = t_3\{t_1t_2 + t_4(t_1 + t_2)\}/\Omega$ , with  $\Omega = t_1t_2 + (t_3 + t_4)(t_1 + t_2)$ . The mean of the distribution of  $y$ , i.e. of the root distribution, is given by

$$\mu = w_1x_1 + w_2x_2 + w_3x_3$$

with  $w_1 = t_2t_3/\Omega$ ,  $w_2 = t_1t_3/\Omega$ , and  $w_3 = \{t_1t_2 + t_4(t_1 + t_2)\}/\Omega$ . These are therefore the Altschul–Carroll–Lipman weights for a tree with three leaves.  $\square$

### Voronoi weights

There are also weighting schemes not based on trees. One approach is based on an image of the sequences from a family lying in 'sequence space'. In general, some will lie in clusters and others will be widely separated. The philosophy of the Voronoi scheme [Sibbald & Argos 1990] is to assume that this unevenness represents effects of sampling, including the 'sampling' performed by natural selection in favouring certain phyla. A more thorough trawl through all eligible sequences of the protein family, or perhaps a multitude of reruns of evolution, should produce a flat distribution within some region. To compensate for the gaps, we want to give sequences a weight proportional to the volume of empty space around them.

If sequence space were two-dimensional, or even low-dimensional, we could use standard methods from computational geometry to divide up space into regions around each example point. The standard approach is to take lines joining neighbouring pairs of points and draw their perpendicular bisectors, extending them till they join up. This produces a partitioning into polygons (in two dimensions) called a *Voronoi diagram* [Preparata & Shamos 1985], which has the property that the polygon around each point is the set of all points closer to that point than any other.

Sequence space is of course a high-dimensional construct in which the Voronoi geometry is hard to picture or calculate. However, we can implement the underlying principle of it by sampling sequences randomly from sequence space and testing to see which of the family sequences each sequence lies closest to. The



trick is in the sampling. This is accomplished by choosing, at each position of the alignment, uniformly from those residues which occur at that position in any sequence. If we count  $n_i$  such sample sequences closest to the  $i$ th family member (dividing up the counts if there is a tie), then we can define the  $i$ th weight to be  $n_i / \sum_k n_k$ .

### Maximum discrimination weights

Another approach to weighting comes indirectly, from focusing initially on a reformulation of the primary goal in building the model [Eddy, Mitchison & Durbin 1995]. Rather than maximising the likelihood of sequences in the family, or even their posterior probability derived from Bayesian priors, we are normally interested in making the correct decision on whether sequences are members of the family or not. We are therefore interested in the probability

$$P(M|x) = \frac{P(x|M)P(M)}{P(x|M)P(M) + P(x|R)(1 - P(M))},$$

where  $x$  is a sequence from the family,  $M$  is the model for the family that we are fitting,  $R$  is our alternative, random model for sequences not in the family, and  $P(M)$  is the prior probability of a new sequence belonging to the family. Given example training sequences  $x^k$ , we would like to maximise the probability of classifying them all correctly, which is

$$D = \prod_k P(M|x^k),$$

not  $\prod P(x^k|M)$  as usual with maximum likelihood based approaches. We call  $D$  the *discrimination* of the model on the set of sequences  $x^k$ . Maximising  $D$  will have the effect of emphasising performance on distant or difficult members of the family. Sequences that are easily classified will have  $P(M|x)$  values very close to one; changing parameters to increase their likelihood  $P(x|M)$  will have very little effect on  $D$ . On the other hand, increasing the likelihood of sequences for which  $P(M|x)$  is small can potentially have a big effect.

It turns out that the parameter values that maximise  $D$  can be shown to be the ones that maximise a weighted version of the likelihood, where the weights are proportional to  $1 - P(M|x_i)$ , i.e. the probability of misclassifying sequence  $i$ . This can be seen from the observation that if  $y = e^x / (K + e^x)$ , then

$$\frac{\partial \log y}{\partial x} = \frac{1}{K + e^x} = K(1 - y).$$

If we set  $x = \log \left( \frac{P(x|M)}{P(x|R)} \right)$ , which is the log likelihood ratio for sequence  $x$ , then  $y = P(M|x)$ . So at a maximum of  $\log D$  we will also be at a maximum of the weighted sum of log likelihood ratios, with weights  $1 - P(M|x_i)$ , and since the

random model is fixed this is equivalent to a maximum of the weighted log likelihood of the model  $M$ . The maximum discrimination criterion therefore amounts to another sequence weighting system.

One difference from previous systems, however, is that these weights are defined in a somewhat circular fashion; they depend upon the model that is being fit. When using maximum discrimination weighting as a method, an iterative approach must be used; an initial set of weights gives rise to a model, from which posterior probabilities  $P(M|x)$  can be calculated, giving rise to new weights, and hence a new model, and so on until convergence is achieved. This iterative re-estimation procedure is analogous to the versions of the EM algorithm used to fit HMM parameters to sets of unlabelled sequences (p. 64 and p. 323).

Maximum discrimination training has a big advantage in that it is directly optimising performance on the type of operation that the model will be used for, ensuring that the most effort is applied to recognising the most distant sequences. On the other hand, exactly the same point can lead to problems. If there is any training sequence that has been misclassified, then the distortion needed to give it a good score can damage performance for correct members of the class. To some extent, though, this same problem occurs with all weighting schemes: incorrectly assigned sequences will be the most distant ones in any tree that gets built from the examples.

### Maximum entropy weights

Finally, we describe two weighting methods based on the idea of trying to make the statistical spread of the model as broad as possible.

Assume column  $i$  of a multiple alignment has  $k_{ia}$  residues of type  $a$  and a total of  $m_i$  different types of residues. To make a distribution as uniform as possible from these counts by weighting each sequence, we can choose a weight for sequence  $k$  of  $1/(m_i k_{ik})$ . Maximum likelihood estimation will then yield a distribution  $p_{ia} = k_{ia}/(m_i k_{ia}) = 1/m_i$ , i.e. all the residues appearing in the column will have the same probability. To illustrate the idea, suppose we have ten sequences with residue A at a site, and one sequence with a B, so the unweighted frequencies of A and B are  $c_A = \frac{10}{11}$ ,  $c_B = \frac{1}{11}$ . The weights of the ten sequences are  $w_1 = w_2 = \dots = w_{10} = 1/(2 \times 10) = 0.05$ , and  $w_{11} = 1/(2 \times 1) = 0.5$ , which have the effect of making the overall weighting for each of A and B equal.

The preceding paragraph only considered one column. With just one weight per sequence, it is of course not possible to make the distribution uniform for all columns in an alignment. However, by averaging over all columns, one may hope to obtain reasonable weights. That is, the weights are calculated as

$$w_k = \frac{1}{\sum_i m_i k_{ik}}$$



and then normalised to sum to one. This weighting scheme was proposed by [Henikoff & Henikoff 1994].

Instead of averaging, there is another approach to combining the information from the different columns that has a simple theoretical justification. A standard measure of the 'uniformity' of a distribution is the entropy (11.8), which is larger the more uniform the distribution is. Indeed, it is easy to see that the weights chosen above based on a single column maximise the entropy of the distribution  $p_{ia}$  for that column. An HMM defines a probability distribution over sequences, and therefore a natural extension of the single column weighting to full sequences is to maximise the entropy of the complete HMM distribution [Krogh & Mitchison 1995]. We will see that, perhaps surprisingly, this is closely related to maximum discrimination weighting.

Let us consider all the sites in an alignment with no gaps. We then sum the entropies from each site, and choose the weights to maximise this sum; that is we maximise  $\sum_i H_i(w \cdot) + \lambda \sum_k w_k$ , where  $H_i(w \cdot) = \sum_a p_{ia} \log p_{ia}$ , and  $p_{ia}$  is the weighted frequency of residue  $a$  at the  $i$ th site, computed as above.

Suppose for instance that we have the sequences  $x^1 = \text{AFA}$ ,  $x^2 = \text{AAC}$ , and  $x^3 = \text{DAC}$ . Giving them weights  $w_1$ ,  $w_2$  and  $w_3$ , respectively, the entropies at each site are

$$H_1(w \cdot) = -(w_1 + w_2) \log(w_1 + w_2) - w_3 \log w_3,$$

$$H_2(w \cdot) = -w_1 \log w_1 - (w_2 + w_3) \log(w_2 + w_3),$$

$$H_3(w \cdot) = -w_1 \log w_1 - (w_2 + w_3) \log(w_2 + w_3).$$

We assume that the weights sum to one, and therefore we have to use a Lagrange multiplier term  $\lambda \sum_k w_k$ , when differentiating and finding the maximum of the entropy. Setting the derivatives of  $H_1(w \cdot) + H_2(w \cdot) + H_3(w \cdot) + \lambda \sum_k w_k$  to zero gives  $(w_1 + w_2)w_1^2 = (w_1 + w_2)(w_2 + w_3)^2 = w_3(w_2 + w_3)^2$ , which implies  $w_1 = w_3 = 0.5$ ,  $w_2 = 0$ . This makes the frequencies in each column equal, which was our goal. If it seems odd to give a sequence zero weight, note that the residue at each site in  $x^2$  is always present in one of the other two sequences. Intuitively,  $x^2$  lies 'between'  $x^1$  and  $x^3$ , (in fact, it would be a possible ancestral sequence of  $x^1$  and  $x^3$  in an evolutionary reconstruction based on parsimony; see Chapter 7).

Another way to view the result of this example is that if we set the model probabilities to be the weighted counts frequencies, as a weighted maximum likelihood procedure would, the resulting model assigns an equal probability to all of the original sequences,  $x^1$ ,  $x^2$  and  $x^3$ . This seems very reasonable, according to the view that all the example sequences should be treated as equally good members of the family for which we are building the model. In fact, Krogh & Mitchison [1995] show that the maximum entropy procedure assigns weights to the example sequences so that some subset of the sequences (perhaps all of them) have non-zero weight and equal probabilities under the resulting model, or they

have a higher probability, in which case they have zero weight. The former can be thought of as boundary points for the region of sequence space occupied by the whole sequence set, while the latter are internal points.

Furthermore, empirical tests indicate that the maximum entropy weights are optimal in the sense that they maximise the minimum score assigned to any of the example sequences [Krogh & Mitchison 1995]. This is an absolute version of the criterion specified in the previous section on maximum discrimination weights; rather than simply weighting the weakest match most strongly, all the parameter-fitting effort is applied to increasing its score, until it reaches that of the other non-zero-weighted sequences. Although satisfying an attractive goal, maximum entropy weighting suffers from the same problems as maximum discrimination: if a sequence is an outlier that should not be a full member of the family, the method will force it in, possibly at a substantial cost in performance on all other sequences. In addition, the rejection of all information from some of the sequences may seem intuitively undesirable.

### Exercise

- 5.6 Compute the weights for the following sequence set, using each of the weighting methods described above except Voronoi weights (which requires random sampling of sequences): AGAA, CCTC, AGTC.

## 5.9 Further reading

PSSM methods were introduced during the 1980s for finding new members of sequence families, although the matrix values were not always obtained using an explicit probability-based derivation. They are also known by other names, such as *weight matrices* [Staden 1988]. More recent papers using related methods include those by Stormo [1990]; Henikoff & Henikoff [1994]; Tatusov, Altschul & Koonin [1994].

The non-probabilistic versions of profiles already have a long history, and many variants of the profile method have been suggested and tested. Thompson, Higgins & Gibson [1994b] and Luthy, Xenarios & Bucher [1994] report an improvement when the sequences are weighted using one of the BLOSUM matrices [Henikoff & Henikoff 1992] instead of a PAM matrix. In Thompson, Higgins & Gibson [1994b] the treatment of gaps is also improved.

Several ways have been suggested for incorporating structural information into profiles. In Luthy, McLachlan & Eisenberg [1991] substitution matrices were estimated for six different structural environments: the three secondary structure elements  $\alpha$ -helix,  $\beta$ -sheet, and 'other' combined with an outside/inside classification, which was based on the exposure of an amino acid to solvent. Other vari-

ations of structural profiles can be found in Bowie, Luthy & Eisenberg [1991]; Wilmanns & Eisenberg [1993].

Early on, profile HMMs were adopted by Baldi *et al.* [1994], who used them to model globins, immunoglobulins and kinases. In this work a different estimation method was also introduced, which was based on gradient descent, see also Baldi & Chauvin [1994]. The same basic structure of profile HMMs has since been used in several different areas. A library of HMMs for all the big protein families has been established under the name of PFAM [Sonnhammer, Eddy & Durbin 1997]. The library of regular expressions called PROSITE [Bairoch, Bucher & Hofmann 1997] is being extended to something essentially like profile HMMs [Bucher *et al.* 1996]. Profile HMMs also have several uses for DNA. For instance they can be used to find DNA repeat family members in large-scale genomic sequence.

## Background on probability

To make our book more self-contained, we have included a last chapter that gathers together the probabilistic ideas and methods we use. The various sections of this chapter are fairly independent, and can be dipped into as the reader wishes. Some parts are more mathematically technical than the rest of the book.

### 11.1 Probability distributions

We introduce here various probability distributions used throughout the book. When the outcomes we wish to assign probabilities to belong to a finite set  $X$ , a probability distribution is simply an assignment of a probability  $p_x$  to each outcome  $x$  in  $X$ . For instance, the probability distribution of outcomes of rolling a fair die would be  $p_x = 1/6$  for the six outcomes  $x = 1, \dots, 6$ .

If we have a continuous variable  $x$ , like the weight of an object, then the probability that that variable takes a specific value, e.g. that the weight is *exactly* 1 pound, is zero. But the probability that  $x$  takes a value in some interval,  $P(x_0 \leq x \leq x_1)$  say, can be well defined and positive. As the width of the interval tends to zero, we may be able to write  $P(x - \delta x/2 \leq x \leq x + \delta x/2) = f(x)\delta x$ , where  $f(x)$  is a function called a *probability density*, or just *density*. The probability of an interval can then be derived by integration:  $P(x_0 \leq x \leq x_1) = \int_{x_0}^{x_1} f(x)dx$ . A density must satisfy  $f(x) \geq 0$ , for all  $x$ , and  $\int_{-\infty}^{\infty} f(x)dx = 1$ . But note that we can have  $f(x) > 1$ . For instance, the density  $f(x) = 10$  for  $0 \leq x \leq 0.1$  and  $f(x) = 0$  elsewhere is well defined.

#### The binomial distribution

The first distribution we consider is perhaps the simplest and most familiar: the *binomial distribution*. It is defined on a finite set consisting of all the possible results of  $N$  tries of an experiment with a binary outcome, '0' or '1'. If  $p$  is the probability of getting a '1' and  $1 - p$  that of getting a '0', the probability that  $k$  out of the  $N$  tries yield a '1' is

$$P(k \text{ '1's out of } N) = \binom{N}{k} p^k (1-p)^{N-k} \quad (11.1)$$

where  $\binom{N}{k}$  denotes the number of ways of choosing  $k$  objects from  $N$ , that is  $N!/((N-k)!k!)$ , and the factorial function is defined for non-negative integers as  $n! = n(n-1)\cdots 1$ , and  $0! = 1$ .

The mean  $m$  and variance  $\sigma^2$  of any distribution  $P$  are defined by  $m = \sum k P(k)$  and  $\sigma^2 = \sum (k-m)^2 P(k)$ . The positive square root of the variance,  $\sigma$ , is called the standard deviation. For the binomial distribution

$$m = \sum_{k=1}^N k \binom{N}{k} p^k (1-p)^{N-k}$$

and

$$\sigma^2 = \sum_{k=1}^N (k-m)^2 \binom{N}{k} p^k (1-p)^{N-k}$$

We can show (Exercise 11.1) that  $m = Np$  and  $\sigma^2 = Np(1-p)$ .

### Exercise

- 11.1 Calculate the mean and variance of the binomial distribution. (Hint: To find  $m$ , differentiate the binomial expansion  $(p+q)^N = \sum_{k=0}^N \binom{N}{k} p^k q^{N-k}$  with respect to  $p$  and set  $q = 1-p$ . For the variance, carry out two differentiations with respect to  $p$ .)

## The Gaussian distribution

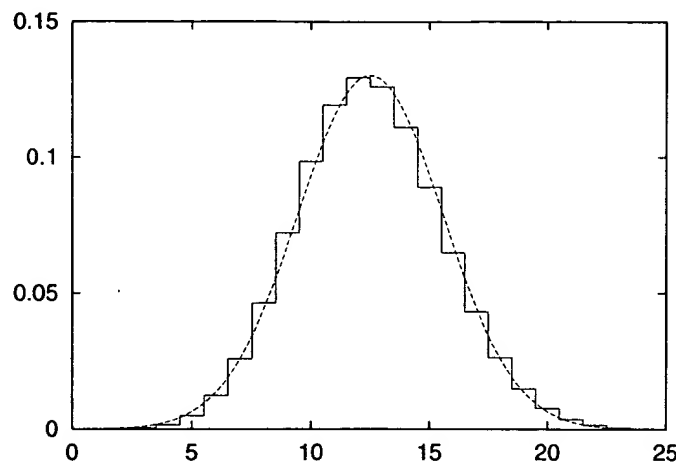
Consider next what happens as we let  $N \rightarrow \infty$ . Both the mean and the variance increase linearly with  $N$ , but we can rescale to give fixed mean and variance, defining the new variable  $u$  by  $u = (k-m)/\sigma = (k-Np)/\sqrt{Np(1-p)}$ . It is a classic result [Keeping 1995] that, in the limit of a large number of events, a binomial distribution becomes a Gaussian (see Figure 11.1), and with the rescaling the density is

$$f(u) = \frac{1}{\sqrt{2\pi}} \exp(-u^2/2). \quad (11.2)$$

This can be regarded as a special case of the central limit theorem, which states that the distribution of a sum of  $N$  independent random variables, normalised to the same mean and variance, tends to a Gaussian as  $N \rightarrow \infty$ . If a single variable takes values '0' or '1' with probabilities  $1-p$  and  $p$ , respectively, the distribution of the sum of  $N$  copies of this is  $P(k) = P(X_1 + \dots + X_N \leq k)$ , and is precisely the binomial considered above.

## The multinomial distribution

The generalisation of the binomial distribution to the case where the experiments have  $K$  independent outcomes with probabilities  $\theta_i$ ,  $i = 1, \dots, K$ , is the *multino-*



**Figure 11.1** The limit for large  $N$  of a binomial tends to a Gaussian. In this case  $N = 40$  and  $p = 1/4$  in (11.1).

*mial distribution.* The probability of getting  $n_i$  occurrences of outcome  $i$  is given by

$$P(n|\theta) = M^{-1}(n) \prod_{i=1}^K \theta_i^{n_i}. \quad (11.3)$$

Here we condition the probability on the parameters  $\theta$  of the distribution, which is a natural thing to do in a Bayesian framework, because then the parameters are themselves random variables. In a classical statistics framework the probability of  $n$  could, for instance, have been denoted by  $P_\theta(n)$ . The normalising constant only depends on the total number of outcomes observed,  $\sum_k n_k$ . For fixed  $\sum_k n_k$  it is

$$M(n) = \frac{n_1! \cdot n_2! \cdots n_K!}{(\sum_k n_k)!} = \frac{\prod_i n_i!}{\sum_k n_k}. \quad (11.4)$$

For  $K = 2$  the multinomial distribution reduces to the binomial distribution.

### Example: Rolling a die

The outcome of rolling a die  $N$  times is described by a multinomial. The probabilities of each of the six outcomes are called  $\theta_1, \dots, \theta_6$ . For a fair die where  $\theta_1 = \dots = \theta_6 = 1/6$  the probability of rolling it a dozen times and getting each outcome twice is

$$\frac{12!}{2!^6} \left(\frac{1}{6}\right)^{12} = 3.4 \times 10^{-3}.$$

□

ere the experiments  
 $K$ , is the *multino-*

### The Dirichlet distribution

In Bayesian statistics we need distributions over probability parameters to use as prior distributions. A natural choice for a density over probabilities is the Dirichlet distribution:

$$\mathcal{D}(\theta|\alpha) = Z^{-1}(\alpha) \prod_{i=1}^K \theta_i^{\alpha_i-1} \delta(\sum_{i=1}^K \theta_i - 1). \quad (11.5)$$

Here  $\alpha = \alpha_1, \dots, \alpha_K$ , with  $\alpha_i > 0$ , are constants specifying the Dirichlet distribution, and the  $\theta_i$  satisfy  $0 \leq \theta_i \leq 1$  and sum to 1, this being indicated by the delta function term  $\delta(\sum_i \theta_i - 1)$ . The algebraic expression for the  $\theta_i$  is the same as for a multinomial distribution. Instead of normalising over the numbers  $n_i$  of occurrences of outcomes, however, we normalise over all possible values of the  $\theta_i$ . To put this another way, the multinomial is a distribution over its exponents  $n_i$ , whereas the Dirichlet is a distribution over the numbers  $\theta_i$  that are exponentiated. The two distributions are said to be conjugate distributions [Casella & Berger 1990], and their close formal relationship leads to a harmonious interplay in many estimation problems.

The normalising factor  $Z$  for the Dirichlet defined in (11.5) can be expressed in terms of the gamma function: [Berger 1985]

$$Z(\alpha) = \int \prod_{i=1}^K \theta_i^{\alpha_i-1} \delta(\sum_i \theta_i - 1) d\theta = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}. \quad (11.6)$$

The gamma function is a generalisation of the factorial function to real values. For integers  $\Gamma(n) = (n-1)!$ . For any positive real number  $x$ ,

$$\Gamma(x+1) = x\Gamma(x). \quad (11.7)$$

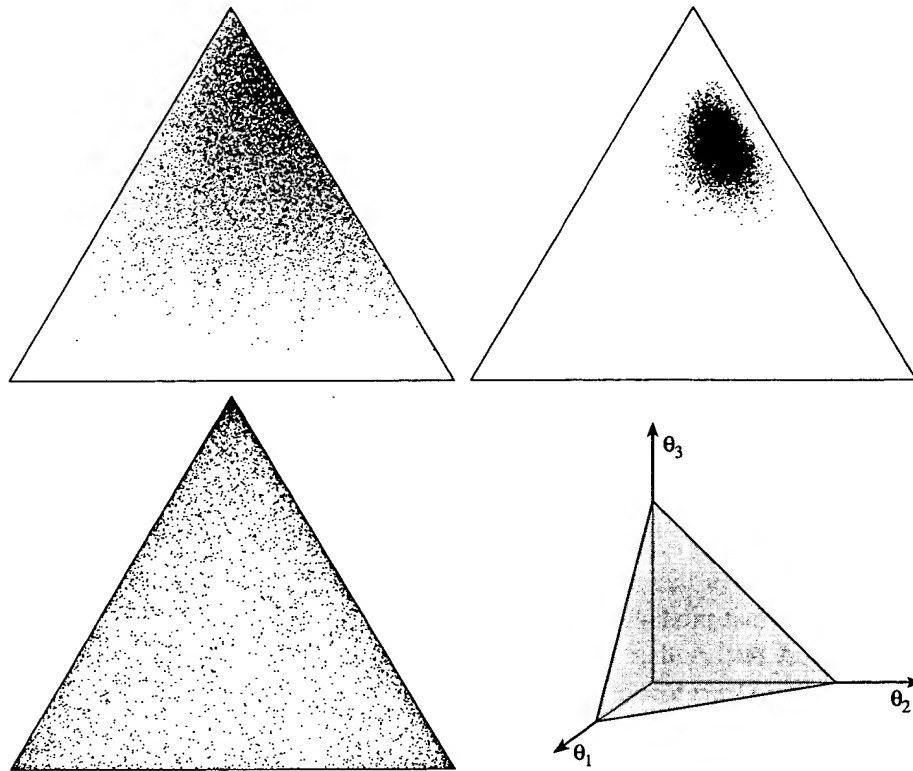
It can be shown that the mean of the Dirichlet distribution is equal to the normalised parameters, i.e. the mean of  $\theta_i$  is  $\alpha_i / \sum_k \alpha_k$ . For instance, the three distributions shown in Figure 11.2 all have the same mean (1/8, 1/4, 5/8), even though the  $\alpha$ s for the top right figure are 10 times larger than those for the top left. Note that larger  $\alpha$ s produce a tighter distribution. Note also that when some  $\alpha_i < 1$  the distribution is peaked at zero for the corresponding  $\theta_i$ , as shown in the bottom left figure.

For two variables ( $K = 2$ ) the Dirichlet distribution reduces to the more widely known beta distribution, and the normalising constant is the beta function.

#### Example: The dice factory

Consider again our example from Chapters 1 and 3 of a probabilistic model of a possibly loaded die with probability parameters  $\theta = \theta_1, \dots, \theta_6$ . Sampling probability vectors  $\theta$  from a Dirichlet parameterised by  $\alpha = \alpha_1, \dots, \alpha_6$  is like a 'dice factory' that produces different dice with different  $\theta$  [MacKay & Peto 1995].



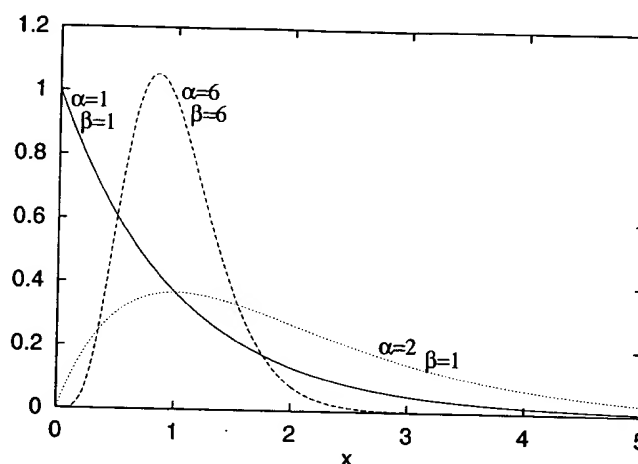


**Figure 11.2** Examples of three-dimensional Dirichlet distributions, each shown by sampling 10000 points, i.e. by choosing points  $\theta$  with probability  $\mathcal{D}(\theta|\alpha)$ . The values of  $\alpha$  used are (1, 2, 5) in the top left figure, (10, 20, 50) top right and (0.1, 0.2, 0.5) bottom left. The probabilities  $\theta$  are displayed as the slice through 3D space  $(\theta_1, \theta_2, \theta_3)$  where  $\sum \theta_i = 1$ ; see the bottom right figure. A point  $(\theta_1, \theta_2, \theta_3)$  is mapped to  $((\theta_2 - \theta_1)/\sqrt{3}, \theta_3)$  in the plane.

Suppose dice factory A has all six  $\alpha_i$  set to 10, and dice factory B has all  $\alpha_i$  set to 2. On average, both factories produce fair dice; the average of  $\theta_i$  is  $\frac{1}{6}$  in both cases. But if we find a loaded die with  $\theta_6 = 0.5, \theta_1 = \dots = \theta_5 = 0.1$ , it is much more likely to have been produced by dice factory B:

$$\begin{aligned}\mathcal{D}(\theta|\alpha_A) &= \frac{\Gamma(60)}{(\Gamma(10))^6} (0.1)^{5(10-1)} (0.5)^{10-1} = 0.119, \\ \mathcal{D}(\theta|\alpha_B) &= \frac{\Gamma(12)}{(\Gamma(2))^6} (0.1)^{5(2-1)} (0.5)^{2-1} = 199.6.\end{aligned}$$

The factory with the higher  $\alpha$  parameters produces a tighter distribution in favour of fair dice. The sum  $\sum \alpha_i$  is inversely proportional to the variance of the Dirichlet. (Don't be alarmed by the Dirichlet density having a value of 199.6; recall that the values of continuous probability densities at any point may be greater than one.)



**Figure 11.3** Gamma distributions  $g(x, \alpha, \beta)$  for  $\alpha = \beta = 1.0$ ,  $\alpha = \beta = 6.0$  and  $\alpha = 2.0$ ,  $\beta = 1.0$ .

A factory that produced almost perfectly fair dice would have very high but equal  $\alpha_i$ . A factory that produced variably unreliable dice that are still fair on average would have low but equal  $\alpha_i$ .  $\square$

### The gamma distribution

The gamma distribution  $g(x, \alpha, \beta)$  is given by

$$g(x, \alpha, \beta) = \frac{e^{-\beta x} x^{\alpha-1} \beta^\alpha}{\Gamma(\alpha)},$$

and is defined for  $0 < x, \alpha, \beta < \infty$ . Its mean is  $\alpha/\beta$  and variance  $\alpha/\beta^2$ .  $\beta$  is simply a scale parameter.

The gamma distribution is conjugate to the Poisson,  $f(n) = e^{-p} p^n / n!$ , which gives the probability of seeing  $n$  events over some interval, when there is a probability  $p$  of an individual event occurring in that interval. Since the number of events in an interval is a rate, the gamma distribution is appropriate for modelling probabilities of rates, just as the Dirichlet is appropriate as a prior for emission probabilities when its conjugate, the multinomial, is used to assign probabilities to counts (p. 319). The gamma distribution has been used to model the rate of evolution at different sites in DNA sequences (p. 215).

### The extreme value distribution

Suppose we take  $N$  samples from the density  $g(x)$ . The probability that the largest amongst them is less than  $x$  is  $G(x)^N$ , where  $G(x) = \int_{-\infty}^x g(u) du$ . The density for the largest value of the set of  $N$  is given by differentiating this with

respect to  $x$ , giving  $Ng(x)G(x)^{N-1}$ . The limit for large  $N$  of  $Ng(x)G(x)^{N-1}$  is called the *extreme value density* (EVD) for  $g(x)$ . It has a wide variety of practical uses, from modelling the breaking-point of a chain (which is determined by the weakest link), to assessing the significance of the maximum score from a set of alignments (see Chapter 2).

Let us compute the EVD when  $g(x)$  is the exponential density  $g(x) = \alpha e^{-\alpha x}$ . Integrating gives  $G(x) = 1 - e^{-\alpha x}$ . Choosing  $y$  so that  $e^{-\alpha y} = 1/N$ , and writing  $z = x - y$ , we find

$$\begin{aligned} Ng(x)G(x)^{N-1} &= N\alpha e^{-\alpha x} (1 - e^{-\alpha x})^{N-1} = \alpha e^{-\alpha z} (1 - e^{-\alpha z}/N)^{N-1} \\ &\rightarrow \alpha e^{-\alpha z} \exp(-e^{-\alpha z}) \text{ for } N \rightarrow \infty, \end{aligned}$$

where we used the well-known limit  $(1 - X/N)^N \rightarrow e^{-X}$  for  $N \rightarrow \infty$ .<sup>1</sup> The cumulative probability (the probability that the extreme value is  $\leq x$ ) is  $\exp(-e^{-\alpha z})$ , and is called a *Gumbel* distribution [Gumbel 1958]. The above density often gives a good approximation to the distribution of extreme values for moderate values of  $N$ . With the exponential density, Figure 11.4 shows that the maximum of a sample of size 10 gives a close approximation to the EVD.

It is a surprising fact that the Gumbel distribution is the EVD for a variety of underlying densities  $g(x)$ ; it holds when  $g(x)$  is a Gaussian too, for instance. More generally, an EVD must have the form  $\exp(-f(a_N x + b_N))$ , where  $a_N$  and  $b_N$  are constants depending on  $N$  and  $f(x)$  is either an exponential  $e^{-x}$  or  $|x|^{-\lambda}$  for some positive constant  $\lambda$  (see Waterman [1995] for a more precise statement of this theorem).

## 11.2 Entropy

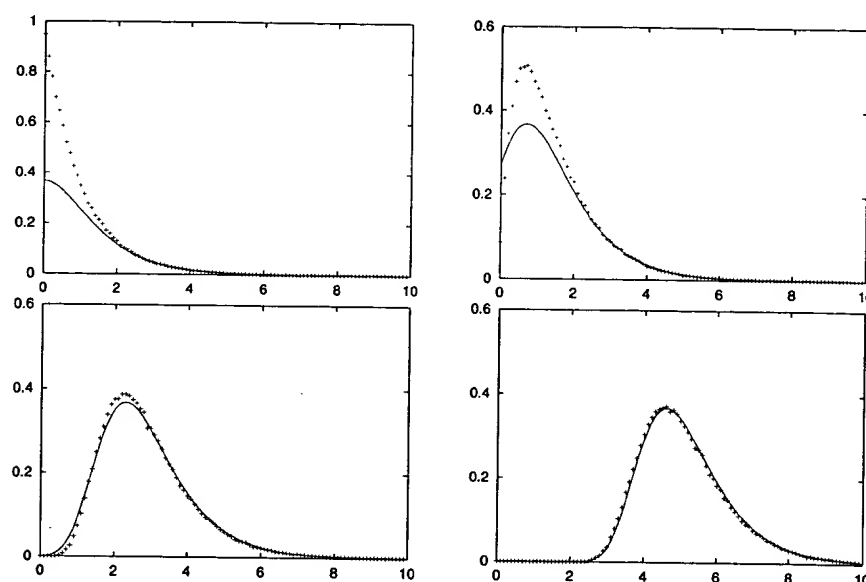
Some of the terminology used in the book is borrowed from information theory (see e.g. Cover & Thomas [1991]). Information theory has strong connections to probabilistic modelling.

An *entropy* is a measure of the average uncertainty of an outcome. Given a random variable  $X$  with probabilities  $P(x_i)$  for discrete set of  $K$  events  $x_1, \dots, x_K$ , the Shannon entropy is defined by

$$H(X) = - \sum_i P(x_i) \log P(x_i). \quad (11.8)$$

In this definition,  $P(x_i) \log P(x_i)$  is taken to be zero if  $P(x_i) = 0$ . Normally we assume that  $\log$  is the natural logarithm (sometimes written  $\ln$ ). However, it is common to use the logarithm base 2 (called  $\log_2$ ), in which case the unit of

<sup>1</sup> There is one delicate point in the above argument. We have to take care that  $e^{-\alpha z}$  cannot grow rapidly with  $N$ , and so invalidate the limit  $(1 - e^{-\alpha z}/N)^N \rightarrow \exp(-e^{-\alpha z})$ . To be more precise, one has to show that the probability of large values of  $e^{-\alpha z}$  according to the distribution  $Ng(x)G(x)^{N-1}$  becomes vanishingly small.



**Figure 11.4** Approximations to the extreme value distribution obtained by sampling  $N$  points from the distribution  $e^{-x}$  on  $0 \leq x < \infty$ , and then taking the maximum. From the top left to bottom right,  $N = 1, 2, 10, 100$ .

entropy is a 'bit'. All logarithms are proportional, e.g.  $\log_2(x) = \log_e(x)/\log_e(2)$ , so theoretically it does not matter which logarithm is used. Often we talk about the entropy of the probability distribution  $P$ ,  $H(P)$ , instead of  $H(X)$ .

The entropy is maximised when all the  $P(x_i)$  are equal ( $P(x_i) = 1/K$ ) and we are maximally uncertain about the outcome of a random sample. The maximum is the  $-\sum_i \frac{1}{K} \log \frac{1}{K} = \log K$ . If we are certain of the outcome of a sample from the distribution, i.e.  $P(x_k) = 1$  for one  $k$  and the other  $P(x_i) = 0$ , the entropy is zero.

Entropy also arises as the expected score of the sequences generated by certain probabilistic models when the score is defined to be the log probability. Suppose, for instance, that the probability of residue  $a$  in some position in a sequence is  $p_a$ . Then there is a probability  $p_a$  of score  $\log p_a$ , and the expected score is  $\sum_a p_a \log p_a$ , namely the negative entropy. The same is true (see Exercise 11.2) when the model defines the probabilities at a set of independent sites.

If you are told the outcome of an event, the uncertainty is reduced from  $H$  to zero, because you have gained information. Therefore entropy is often equated with information. This can be confusing; it leads to the quite counterintuitive view that the more random something is (the higher the entropy), the more information it has. It is not confusing if we think of information as a difference in entropy. More generally, *information content* or just *information* is a measure of a reduction in uncertainty after some 'message' is received; hence, the difference

between the entropy before and the entropy after the message:

$$I(X) = H_{\text{before}} - H_{\text{after}}. \quad (11.9)$$

The uncertainty is not always reduced to zero; there may be noise on the communications channel, for instance, and we may remain somewhat uncertain of the outcome, in which case  $H_{\text{after}}$  is positive and the information is less than the original entropy.

In information theory it is often assumed that the probability distributions are known exactly. In many applications, however, the true distributions are not known, and therefore entropies are calculated from the frequencies of events rather than the true distributions; see Examples below.

### Example: Entropy of random DNA

If each symbol (A, C, G, or T) of a DNA sequence occurs equiprobably ( $p_a = 1/4$ ) then the entropy per DNA symbol is  $-\sum_a p_a \log_2 p_a = 2$  bits.

We can think of the entropy as the number of binary yes/no questions needed to discover the outcome. For example, for random DNA, we need two questions: 'purine or pyrimidine?' followed by 'A or G?' if the answer is 'purine', and 'C or T?' otherwise.  $\square$

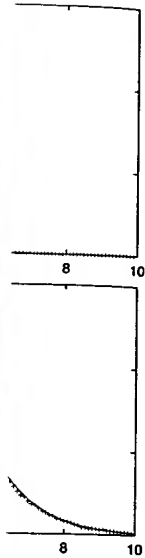
### Example: Information content of a conserved position

Information content can be used to measure the degree of conservation at a site in a DNA or protein sequence alignment. Say we expect a DNA sequence to be random ( $p_a = 0.25$ ;  $H_{\text{before}} = 2$  bits), but we observe that a particular position in a number of related sequences is always an A or a G with  $p_A = 0.7$  and  $p_G = 0.3$ . Thus  $H_{\text{after}} = -0.7 \log_2 0.7 - 0.3 \log_2 0.3 = 0.88$  bits. The information content of this position is said to be  $2 - 0.88 = 1.12$  bits. The more conserved the position, the higher the information content.

Notice, however, that the information content can be negative if the observed distribution has a higher entropy (is more 'random') than expected. For finding unusual patterns it is therefore better to measure the difference between the distributions by the relative entropy described below.  $\square$

### Exercise

11.2 Assume a model in which  $p_i(a)$  is the probability of amino acid  $a$  occurring in the  $i$ th position of a sequence of length  $l$ . The amino acids are considered independent. What is the probability  $P(x)$  of a particular sequence  $x = x_1, \dots, x_l$ ? Show that the average of the log of the probability is the negative entropy  $\sum P(x) \log P(x)$ , where the sum is over all possible sequences  $x$  of length  $l$ .



ained by  
en taking

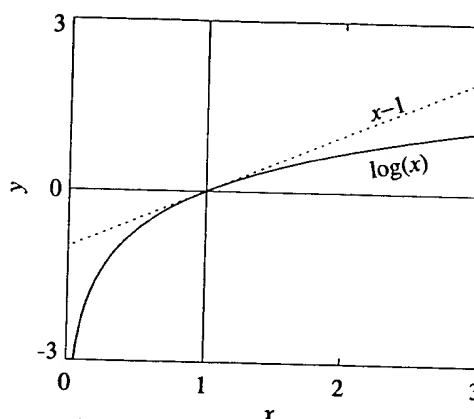
$\log_e(x)/\log_e(2)$ ,  
we talk about  
( $X$ ).

$1/K$ ) and we  
the maximum  
a sample from  
the entropy is

ited by certain  
ility. Suppose,  
a sequence is  
ected score is  
Exercise 11.2)

es.

ced from  $H$  to  
often equated  
ounterintuitive  
, the more in-  
a difference in  
s a measure of  
the difference



**Figure 11.5** Proof that the relative entropy (11.10) is always positive or zero if  $P(x_i) = Q(x_i)$  for all  $i$ . From this graph it can be seen that  $\log(x) \leq x - 1$  with equality only if  $x = 1$ . It follows that  $-H(P||Q) = \sum_i P(x_i) \log(Q(x_i)/P(x_i)) \leq \sum_i P(x_i)(Q(x_i)/P(x_i) - 1) = 0$ , with equality holding only if, for each  $i$ ,  $Q(x_i) = P(x_i)$ .

### Relative entropy and mutual information

We return to the definition of different types of entropy. For two distributions  $P$  and  $Q$  the *relative entropy* (also known as the Kullback–Leibler ‘distance’) is defined by

$$H(P||Q) = \sum_i P(x_i) \log \frac{P(x_i)}{Q(x_i)}. \quad (11.10)$$

Information content and relative entropy are the same if the  $Q$  is a uniform ‘background distribution’ ( $Q(x_i) = \frac{1}{K}$ ) that represents a completely naive initial state for  $H_{\text{before}}$ . The two terms are sometimes used interchangeably.

Relative entropy has the property that it is always greater than or equal to zero. It is easy to show that  $H(P||Q) \geq 0$  with equality if and only if  $P(x_i) = Q(x_i)$  for all  $i$  (see Figure 11.5). It is often useful to think of the relative entropy  $H(P||Q)$  as a distance between the probability distributions  $P$  and  $Q$ . However, it is not symmetric,  $H(P||Q) \neq H(Q||P)$ , and it does not fulfil the formal requirements of a proper mathematical distance measure.

The relative entropy often arises as the expected score in models where the score is defined as the *log-odds*, i.e.  $P(\text{data}|M)/P(\text{data}|R)$ , where  $M$  is the model, and  $R$  is a null model. If  $p_a$  is the probability of residue  $a$  in some position in a sequence according to  $M$ , and  $q_a$  its probability according to  $R$ , then the score for residue  $a$  is  $\log(p_a/q_a)$ , and the expected score is  $\sum_a p_a \log(p_a/q_a)$ , which is the relative entropy.

Another important entropy measure is the *mutual information*. Two random variables  $X$  and  $Y$  are independent if  $P(X, Y) = P(X)P(Y)$ . It is interesting to

know how independent they are, and that can be measured by the relative entropy 'distance' between the distributions  $P(X, Y)$  and  $P(X)P(Y)$ ,

$$M(X; Y) = \sum_{i,j} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}, \quad (11.11)$$

where the possible values for  $X$  and  $Y$  are  $\{x_i\}$  and  $\{y_j\}$ . This is the mutual information.  $M(X; Y)$  can be interpreted as the amount of information that we acquire about outcome  $X$  when we are told outcome  $Y$ .

The mutual information is maximal when  $X$  and  $Y$  always covary. If for instance all pairs except AT, TA, GC, and CG have probability zero for two positions  $i$  and  $j$  in some aligned DNA sequences, there is maximal covariation. For this situation we will always have  $P(x_i, y_j) = P(x_i) = P(y_j)$  or  $P(x_i, y_j) = 0$ , and therefore  $M = -\sum_i P(x_i) \log P(x_i)$ . This is the entropy of  $X$  (or  $Y$ ), so it is maximal for a uniform distribution, and the maximum is  $\log K$  (assuming that  $X$  and  $Y$  have the same number,  $K$ , of possible outcomes). The maximum mutual information for DNA sequences is therefore  $\log_2 4 = 2$  bits.

In Figure 10.6 the mutual information (calculated from frequencies) between every pair of columns in an RNA alignment is shown.

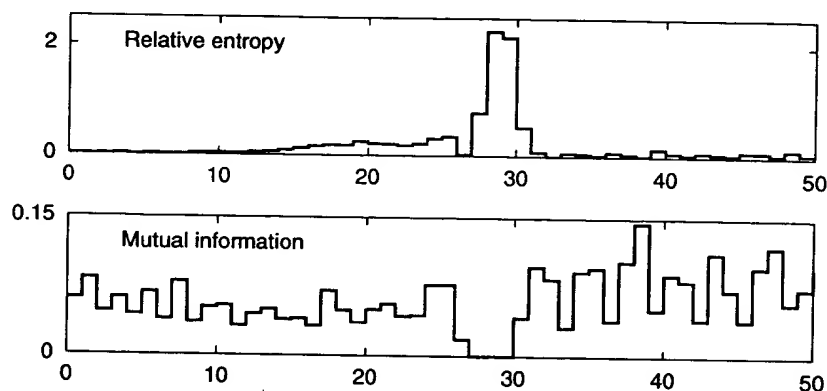
#### Example: Acceptor sites

Relative entropy is useful for finding unusual patterns in biological sequences. To illustrate this we extracted 757 acceptor sites from a database with human genes. The acceptor site is the splice site at the 3' end of the intron where the intron is spliced out to make the messenger RNA. The last two bases of the intron are almost always AG, and in this dataset they all are. We only took acceptor sites of introns occurring between two codons, i.e. not splicing in the middle of a codon. We extracted 30 bases upstream of the splice site and 20 bases downstream. In Figure 11.6 you see a small arbitrary sample of the sequences.

At each position  $i$  the frequency  $p_i(a)$  of the four nucleotides was found, and the relative entropy  $\sum_a p_i(a) \log_2 [p_i(a)/q_a]$  calculated, where  $q_a$  is the overall distribution of the four nucleotides in the sequences. We plot this in Figure 11.6. At the AG consensus the relative entropy is very high (equal to  $-\log_2(q_A)$  and  $-\log_2(q_G)$  respectively). There is an interesting structure in the relative entropy upstream of the site with a minimum just two bases before the AG. There is a weak periodic signal (barely visible) of the relative entropy in the coding region, which is due to the different base composition in the three reading frames. See Brunak, Engelbrecht & Knudsen [1991] and Hebsgaard *et al.* [1996] for more discussion of information in splice sites, and Schneider & Stephens [1990] for colourful ways of displaying various entropy measures.

To see if the neighbouring positions are independent, the mutual information between the columns was calculated. For two neighbouring columns (say  $i$  and  $i + 1$ ) the frequency of pairs  $p_i(a, b)$  was found by counting how many times





## Example sequences

```

CTTCTCAAATAACTGTGCCTCTCCCTCCAGATTCTCAACCTAACAACCTGA
CTGCTCACCGACGAACGACATTTTCCACAGGAGCCGACCTGCCTACAGAC
GGTTCCCTCTTGGCTTCCATGTCTGACAGGTGGATGAAGACTACATCCA
ACTAACTCTCCTCCTCGTGTGTCTCCCCAGCCCGTGTCCAGCCCACCCA
TTGATAACATGACATTTTCTTTTCTACAGAATGAAACAGTAGAAGTCAT
TCTACCGTCCCTTTCCACACACTCTGCAGAAGGTGGTGTGTCTTCTGG
CTTTTTTCTCTCCTATGTGCATCCCCCAGGAGCTGGCTGAATATGAATA
GCTAATAGCTTGCTTATTTATTTAATAGAGGGCTTCCGTTACAAGATGAG
AATTTAGTTTATTCCCATGTGACCTGCAGGTAAATGAAGAAGGCAGTGA
ACTCTGCTCACTGTCACTTTGCTCCACAGCGTCCGCTGTGCAATGGCAG
ACCTCCTAACGTTGTTGGGTTTCTTTCAGAACTTTGCTGCCAGATGGC
GTAAACCCCTCATTTTCTGTTCCGATGCAGGGCCCATGGGACCTCGAGG
AGAAGTGACATTTTCTTATATGTTGACAGGGTGGTGACTTCACACGCCA
CTGGTGTGAGGACCTGCCCTCTCTTTCAAGGGTGAACCTGGTATTGCTGG
ACCTTGGGCACTGTGTTCCCTTGTCTTAGCACTGGCAGATCCCCCTGAG
TTTTGTTATGCAATTATTGTTTCTTACAGGGCCCTCTACTAAAGAAGGA
GCATCACCTGTGAGCTCCCTGTGTCCACAGGCTCTGCAGCGGCTCAGGA

```

**Figure 11.6** Plots of relative entropy and mutual information for acceptor sites. Below is shown a sample of the sequences. Note the peak in relative entropy and dip in mutual information at the conserved AG.

$a$  occurred in column  $i$  and  $b$  occurred in column  $i + 1$ . From this the mutual information  $\sum_{a,b} p_i(a,b) \log_2[p_i(a,b)/p_i(a)p_{i+1}(b)]$  was calculated, and is also plotted in Figure 11.6.

Notice that the mutual information is zero at the AG consensus: knowing that the first is A conveys no information about the next position, because it is always a G. The mutual information around the acceptor site is much less than the maximum of 2 bits, but it is non-zero, and it shows that there are correlations between neighbouring positions. This is true in most DNA. A clear periodic pattern is seen for the coding region, showing that the nucleotides are dependent in the three reading frames. □

## Exercises

- 11.3 Prove the above assertion about the equivalence of information content and relative entropy when  $q$  is uniform.
- 11.4 Show that  $M(X; Y) = M(Y; X)$ .
- 11.5 Show that  $M(X; Y) = H(X) + H(Y) - H(Y, X)$ , where  $H(Y, X)$  is the entropy of the joint distribution  $P(X, Y)$ .

## 11.3 Inference

Probabilistic models are the main focus of this book. A model can be anything from a simple distribution to a complex stochastic grammar with many implicit probability distributions. Once the type of model is chosen, the parameters of the model have to be *inferred* from data. For instance, we may model the outcome of rolling a die with a multinomial distribution. Suppose the number of observations yielding  $i$  is  $n_i$  ( $i = 1, \dots, 6$ ). We do not know if it is a fair die, so we need to estimate the parameters of the multinomial distribution, i.e. the probability  $\theta_i$  of getting  $i$  in a throw of the die. Here, we consider the different strategies that might be used for inference in general. For more background, see Ripley [1996] and MacKay [1992].

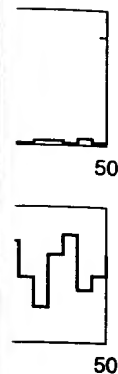
## Maximum likelihood

Let us suppose, then, that we wish to infer parameters  $\theta = \{\theta_i\}$  for a model  $M$  from a set of data  $D$ . The most obvious strategy is to maximise  $P(D|\theta, M)$  over all possible  $\theta$ . This is called the *maximum likelihood* criterion. Formally we write

$$\theta^{\text{ML}} = \underset{\theta}{\operatorname{argmax}} P(D|\theta, M). \quad (11.12)$$

Generally speaking, when we treat  $P(x|y)$  as a function of  $x$  we refer to it as a probability; when we treat it as a function of  $y$  we call it a likelihood. Note that a likelihood is not a probability distribution or density, but simply a function of the variable  $y$ .

Maximum likelihood has some desirable properties. For instance, it is *consistent*, in the sense that the parameter value  $\theta_0$  used to generate the dataset will also, in the limit of a large amount of data, be the value that maximises the likelihood. To see this, suppose there are  $K$  observable outcomes  $\omega_1, \dots, \omega_K$  of the model  $M$  (e.g. the  $4^n$  possible assignments of nucleotides at a site in an aligned set of sequences). Then the frequency  $n_i / \sum n_k$  of occurrences of  $\omega_i$  will tend to  $P(\omega_i|\theta_0, M)$  as the amount of data increases (see Exercise 11.6). Hence the log likelihood for parameter  $\theta$ , given by  $\sum_i (n_i / \sum n_k) \log P(\omega_i|\theta, M)$



AACTGA  
ACAGAC  
ACATCCA  
ACACCCA  
AGTCAT  
ATTCTGG  
ATGAATA  
AGATGAG  
ACAGTGA  
ATGGCAG  
AGATGGC  
ATCGAGG  
ACGCCA  
ATTGCTGG  
ACCTGAG  
AGAAGGA  
TCAGGGA

acceptor  
relative

his the mutual  
ed, and is also

knowing that  
ecause it is al-  
h less than the  
orrelations be-  
eriodic pattern  
pendent in the

tends to  $\sum_i P(\omega_i|\theta_0, M) \log P(\omega_i|\theta, M)$ . The positivity of relative entropy implies that  $\sum_i P(\omega_i|\theta_0, M) \log P(\omega_i|\theta_0, M) \geq \sum_i P(\omega_i|\theta_0, M) \log P(\omega_i|\theta, M)$ , for all  $\theta$ . Thus the likelihood is maximised by  $\theta_0$ .

A drawback of maximum likelihood is that it can give poor results when the data are scanty; we would be wiser then to rely on more prior knowledge. Consider the dice example and assume we want to estimate the multinomial parameters from, say, three different rolls of the dice. It is shown on p. 319 that the maximum likelihood estimate of  $\theta_i$  is  $n_i / \sum n_k$ , i.e. it is 0 for at least three of the parameters. This is obviously a bad estimate for most dice, and we would like a way to incorporate the prior knowledge that we expect all the parameters to be quite close to 1/6.

### Exercise

- 11.6 The *weak law of large numbers* says that the mean of a sample of size  $N$  differs from the true mean by an amount  $d$  or more with probability  $\sigma^2/(Nd^2)$ , where  $\sigma^2$  is the variance of the distribution. Show that this implies that  $n_i / \sum n_k$  tends to  $P(\omega_i)$  as  $\sum n_k \rightarrow \infty$ , where  $n_i$  is the frequency of occurrence of  $\omega_i$ .

### The posterior probability distribution

The way to introduce prior knowledge is to use Bayes' theorem. Suppose there is a probability distribution over the parameters  $\theta$ . Conditioning throughout on  $M$  gives the following version of Bayes' theorem:

$$P(\theta|D, M) = \frac{P(D|\theta, M)P(\theta|M)}{P(D|M)}. \quad (11.13)$$

The prior  $P(\theta|M)$  has to be chosen in some reasonable manner, and that is the art of Bayesian estimation. This freedom to choose a prior has made Bayesian statistics controversial at times, but we believe it is a very convenient framework for incorporating prior (biological) knowledge into statistical estimation.

$P(\theta|D, M)$  is the posterior probability for the parameters, given the data and the model. The posterior can be used for inference in various ways. We can sample from it (see Section 11.4), and thereby locate regions of high probability for the model parameters. In Section 8.4 we show how this can be done for probabilistic models of phylogeny. If we want a specific set of parameter values for the model, we might be guided by analogy with ML and choose the *maximum a posteriori* probability (MAP) estimate,

$$\theta^{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} P(D|\theta, M)P(\theta|M). \quad (11.14)$$

Note that we ignore the data prior  $P(D|M)$ , because it does not depend on the

parameters  $\theta$  and thus the maximum point  $\theta^{\text{MAP}}$  is independent of it. Another possibility is to take the *posterior mean* estimator (PME), which chooses the average of all parameter sets weighted by the posterior:

$$\theta^{\text{PME}} = \int \theta P(\theta|n) d\theta. \quad (11.15)$$

The integral is over all valid probability vectors, i.e. all those that sum to one. In the following we will derive the PME for a multinomial distribution with a certain prior.

Both MAP and PME estimators are considered a little suspicious, because a non-linear transformation of the parameters usually changes the result. In technical terms they are not *equivariant* [Ripley 1996]. To see what's going on, we need to consider the effects of change of variables on densities.

### Change of variables

Given a density  $f(x)$ , suppose there is a change of variable  $x = \phi(y)$ . Then we can define a density  $g(y)$  by  $g(y) = f(\phi(y)) |\phi'(y)|$ . The derivative of  $\phi$ ,  $\phi'(y)$ , is there because the interval  $\delta x$  corresponds to an interval  $\delta y \phi'(y)$  under the transform  $\phi$ , so the amount of the  $f$  density that is swept out under  $\phi$  is proportional to this derivative; taking the derivative's absolute value ensures that the density is positive. This definition produces a correctly normalised density because  $\int g(y) dy = \int f(\phi(y)) |\phi'(y)| dy = \int f(x) dx = 1$ ,  $f$  being a density. We write the transformation rule formally as

$$g(y) = f(\phi(y)) |\phi'(y)|. \quad (11.16)$$

The function  $f(\phi(y))$  clearly has the same maximum as  $f(x)$ . When we multiply by  $|\phi'(y)|$ , however, this maximum may shift (see Exercise 11.7). Now, the posterior  $P(\theta|D, M)$  is a density, so the peak chosen by MAP can likewise change under a transformation. A similar argument shows that the PME can change under a coordinate transformation.

In contrast, the likelihood  $P(D|\theta, M)$  does not transform like a density; it is simply a function of  $\theta$  and a change of coordinates leaves the peak unchanged, just as the peak of  $f(\phi(y))$  remains the same as that of  $f(x)$  [Edwards 1992].

### Exercise

- 11.7 Let  $f(x) = 2(1-x)$  be a density on  $[0, 1]$ . Show how this transforms to a density on  $y$  under  $x = y^2$ . Show that the peak and the PME of the density both shift under this transformation.

## 11.4 Sampling

Given probabilities  $P(x_i)$  defined on the members  $x_i$  of a finite set  $X$ , to *sample* from this set means to pick elements  $x_i$  randomly with probability  $P(x_i)$ .

The basic practical tool for sampling is a function derived from a computer's pseudo-random number generator (i.e. the function called `rand[]`, or something similar), that picks numbers randomly from the interval  $[0, 1]$  with the uniform density. Let us call this function `rand[0, 1]`. Using it, we can choose elements  $x_i$  with frequency  $P(x_i)$ . We set  $y = \text{rand}[0, 1]$ , and then choose our element  $x_i$  by finding that  $i$  for which  $P(x_1) + \dots + P(x_{i-1}) < \text{rand}[0, 1] < P(x_1) + \dots + P(x_{i-1}) + P(x_i)$ . Clearly, the probability of `rand[]` lying in this range is  $P(x_i)$ , so  $x_i$  is picked with the correct probability.

It is actually not easy to produce random numbers with a computer. The standard function for pseudo-random numbers is usually very primitive, and will not be good enough for some applications. For example, the standard `rand[]` function on many UNIX computers returns an integer between 0 and  $2^{15} - 1$ , and one would expect to obtain 'random' bits (0 or 1) with this function by taking the value returned modulo 2. However, this gives a sequence where 0 and 1 alternate, which is clearly not random at all. On most systems there are other (and better) functions to choose from. See for instance Press *et al.* [1992] for a discussion of random number generators.

### Sampling by transformation from a uniform distribution

The concept of sampling applies also to densities: Given a density  $f$ , to sample from it is to pick elements  $x$  from the space on which  $f$  is defined so that the probability of picking a point in an arbitrarily small region  $\delta R$  round the point  $x$  is  $f(x)\delta R$ . Sampling of densities can be accomplished by using pseudo-random numbers that sample from the uniform density on  $[0, 1]$ , and applying a change of variables that changes the density appropriately.

The theory of this goes as follows: Suppose we are given a density  $f(x)$ , and a map  $x = \phi(y)$ . From (11.16) we know that  $g(y) = f(\phi(y))\phi'(y)$ . If  $f$  is uniform, we have  $g(y) = \phi'(y)$ , so  $\phi$  can be obtained by integration,  $\phi(y) = \int_b^y g(u)du$ , where  $b$  is some suitable lower bound. However, we want to pick points in  $x$  using a good pseudo-random number generator, and then map them to  $y$ . For this, we require the inverse function to  $\phi$ , namely  $y = \phi^{-1}(x)$ .

Suppose for instance that we want to sample from a Gaussian. We define the cumulative Gaussian map  $\phi(y) = \int_{-\infty}^y e^{-u^2/2}/\sqrt{2\pi} du$ , and let  $y = \phi^{-1}(x)$ . We could make a look-up table to evaluate the inverse cumulative Gaussian function, but this is rather clumsy, and some other approach may be more convenient (e.g. Exercise 11.10).

The transformation method also applies more generally to functions of  $K$  vari-

ables, but then (11.16) must be replaced by

$$g(y_1, \dots, y_K) = f(\phi_1(y_1, \dots, y_K), \dots, \phi_K(y_1, \dots, y_K)) |J(\phi)|, \quad (11.17)$$

where  $J(\phi)$  is the Jacobian, whose  $(i, j)$ -th entry is  $\partial \phi_i / \partial y_j$  [Feller 1971].

### Exercises

- 11.8 Show that the function  $g(y) = \alpha^\lambda \lambda y^{\lambda-1} / (\alpha^\lambda + y^\lambda)^2$  is a density on  $0 \leq y < \infty$ . Show that picking  $x$  uniformly from  $(0, 1)$  and mapping  $x$  to  $y = \alpha(\frac{x}{1-x})^{1/\lambda}$  samples from  $g(y)$ .
- 11.9 Define a mapping  $\phi$  from the variables  $(x, y)$  to  $(u, w)$  by  $x = uw$ ,  $y = (1-u)w$ . Show that  $J(\phi) = w$ , where  $J$  is the Jacobian.
- 11.10 (Calculus needed!) Suppose we pick two random numbers  $x$  and  $y$  in the range  $[0, 1]$  and map  $(x, y)$  to the sample point  $\cos(2\pi x) \log(1/y^2)$ . Prove that this samples correctly from a Gaussian. This is called the Box-Muller method [Press *et al.* 1992].

### Sampling from a Dirichlet by rejection

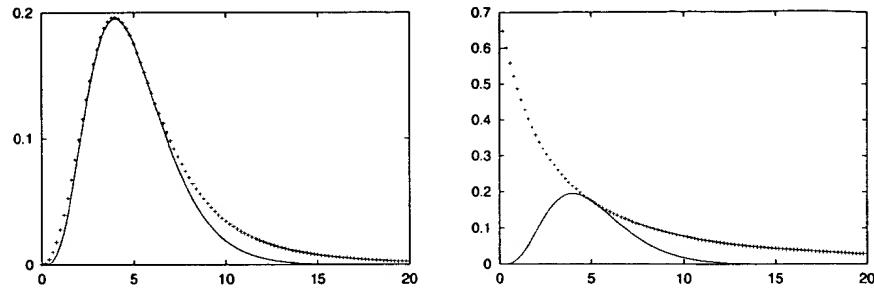
We consider now the problem of sampling from a Dirichlet, which illustrates some important principles. Suppose first that we can sample from the gamma distribution  $g(x, \alpha, 1)$

$$g(x, \alpha, 1) = e^{-x} x^{\alpha-1} / \Gamma(\alpha)$$

for  $0 < x < \infty$  (see p. 304). If we take sampled values  $x_1$  and  $x_2$  from two gamma distributions with parameters  $\alpha_1$  and  $\alpha_2$ , respectively, then we can define a pair  $(u, v)$  with  $u + v = 1$ , by setting  $u = x_1 / (x_1 + x_2)$ ,  $v = x_2 / (x_1 + x_2)$ ; equivalently, we can set  $x_1 = uw$ ,  $x_2 = (1-u)w$  and integrate over  $w$ . Using (11.17) and the results of Exercise 11.9, the distribution  $D(u, v)$  of pairs  $(u, v)$  is given by

$$\begin{aligned} D(u, v) &= \frac{\int_0^\infty \delta(u+v-1) e^{-uw} (uw)^{\alpha_1-1} e^{-vw} (vw)^{\alpha_2-1} w dw}{\Gamma(\alpha_1) \Gamma(\alpha_2)} \\ &= \frac{u^{\alpha_1-1} v^{\alpha_2-1} \delta(u+v-1)}{\Gamma(\alpha_1) \Gamma(\alpha_2)} \int_0^\infty e^{-w} w^{\alpha_1+\alpha_2-1} dw \\ &= u^{\alpha_1-1} v^{\alpha_2-1} \delta(u+v-1) \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \Gamma(\alpha_2)} \\ &= \mathcal{D}(u, v | \alpha_1, \alpha_2), \end{aligned} \quad (11.18)$$

where  $\mathcal{D}(u, v | \alpha_1, \alpha_2)$  is the Dirichlet distribution with parameters  $\alpha_1, \alpha_2$ . In other words, to sample from a Dirichlet distribution of two variables (a beta distribution), we sample from two gamma distributions, whose exponents are those of the components of the Dirichlet in question, and then normalise the sampled numbers to give probabilities. This elegant result extends to Dirichlets of any number of variables (Exercise 11.11).



**Figure 11.7** *Rejection sampling:* We wish to sample from a gamma distribution  $g(x, \alpha, 1)$  (continuous line). It is possible to sample from the function  $f$  given by (11.19) ('+' signs), whose value always exceeds that of the gamma distribution. Having sampled a point  $x$  from  $f$ , this point is accepted with a probability equal to the ratio of the gamma distribution and  $f$  at that point, i.e. with probability  $g(x, \alpha, 1)/f(x)$ . The left figure shows  $f$  with  $\alpha = 5$ ,  $\lambda = 3$ , the right with  $\alpha = 5$ ,  $\lambda = 1$ .

We can sample from a Dirichlet, therefore, if we know how to sample from a gamma distribution. Now we can show (Exercise 11.12) that  $g(x, \alpha, 1) \leq f(x)$ , where

$$f(x) = \frac{4e^{-\alpha} \alpha^{\lambda+\alpha} x^{\lambda-1}}{\Gamma(\alpha)(\alpha^{\lambda} + x^{\lambda})^2}, \quad (11.19)$$

and  $\lambda = \sqrt{2\alpha - 1}$ . It follows that, if  $\text{rand}[0, 1]$  truly samples uniformly between 0 and 1, then  $P(\text{rand}[0, 1] < g(x, \alpha, 1)/f(x)) = g(x, \alpha, 1)/f(x)$ . Thus if we first sample from the distribution  $f$ , picking a point  $x$  with probability  $f(x)$ , and accept  $x$  if  $\text{rand}[0, 1] < g(x, \alpha, 1)/f(x)$ , then

$$P(x) = f(x)P(\text{rand}[0, 1] < g(x, \alpha, 1)/f(x)) = g(x, \alpha, 1).$$

So this two-stage procedure enables us to sample from the gamma distribution. It remains only to show how to sample from  $f$ . But Exercise 11.8 shows that choosing  $u$  from  $[0, 1]$  by  $\text{rand}[0, 1]$  and defining  $x = \alpha(u/1 - u)^{1/\lambda}$  is equivalent to sampling from  $f$ . For more details of the material in this section, and also for the appropriate procedure in the case where  $0 < \alpha < 1$ , see Law & Kelton [1991]. Figure 11.2 was generated using this method.

This is an example of *rejection sampling*, the distribution  $g$  being obtained by 'trimming down' from the distribution  $f$ , which is analytically tractable and always larger than  $g$ . This only works well if  $f(x)$  is a good approximation to  $g(x, \alpha, 1)$ ; if it is not, the rejection rate will be high. The function  $f$  gives a good approximation to  $g(x, \alpha, 1)$  over the range where both functions are large, i.e. where they will be most frequently sampled from. The choice of  $\lambda$  is in fact optimal for this purpose. For instance, with  $\alpha = 5$  and  $\lambda = \sqrt{2\alpha - 1} = 3$ ,



only 14% of points are rejected (Figure 11.7, left figure), whereas with  $\lambda = 1$  (Figure 11.7, right figure), 65% are rejected.

### Exercises

- 11.11 Show that (11.18) can be extended to the case of  $K$  gamma distributions, i.e. that sampling from  $g(x, \alpha_i, 1)$ , for  $i = 1, \dots, K$ , then averaging, is equivalent to sampling from the Dirichlet  $D(\theta_1, \dots, \theta_K | \alpha_1, \dots, \alpha_K)$ . (Hint: Show that the Jacobian of the map  $x_i = u_i w$ , for  $i \leq K-1$ , and  $x_K = (1 - \sum u_i)w$  is equal to  $w^{K-1}$ .)
- 11.12 Prove that  $g(x, \alpha, 1) \leq f(x)$ , for all  $x$  and  $\alpha > 1$  and  $1 \geq \lambda \leq \sqrt{2\alpha - 1}$ , where  $f(x)$  is defined by (11.19). What happens when  $\lambda > \sqrt{2\alpha - 1}$ ?

### Sampling with the Metropolis algorithm

We often want to sample from a probabilistic model, where the analytic methods that underlie the transformation method or rejection sampling are not available. One possible approach then is to use a Markov chain defined on the space  $X$  of outcomes [Neal 1996]. We assume here that  $X$  is finite, although the ideas carry over to continuous variables and densities.

Given a point  $x$ , a chain specifies a probability  $\tau(y|x)$  for the transition  $x \rightarrow y$  to a point  $y$ . If we can sample from the distribution  $\tau(y|x)$ , i.e. given  $x$  can pick a  $y$  with probability  $\tau(y|x)$ , then we can generate a sequence  $\{y_i\}$  where each  $y_i$  is picked by sampling from the distribution  $\tau(y|y_{i-1})$ .

Suppose now that we can find a  $\tau$  satisfying

$$P(x)\tau(y|x) = P(y)\tau(x|y). \quad (11.20)$$

This is called the condition of *detailed balance*. It turns out that detailed balance implies

$$\frac{1}{N} \lim_{N \rightarrow \infty} C(y_i = x) = P(x), \quad (11.21)$$

for all points  $x$ , where  $C(y_i = x)$  is the number of times  $y_i = x$  in the sequence of length  $N$ . We can therefore approximate  $P$  as closely as we like by taking long enough sequences of  $\{y_i\}$  sampled using  $\tau$ . This statement needs to be qualified: Clearly, the chain needs to be able to reach every point  $y$  from any other point  $x$ ; in other words, there must be a sequence of transitions that can go from  $x$  to  $y$ , for any  $x$  and  $y$ .

If we have a transition process  $\tau$  that satisfies (11.20), therefore, the sequences it generates will sample  $P$  correctly. But can we find such a process? A method that achieves this is the Metropolis algorithm. It has two parts:

- (1) A symmetric *proposal* mechanism. Given a point  $x$ , this selects a point  $y$  with probability  $F(y|x)$ . *Symmetry* means that  $F(y|x) = F(x|y)$ .

(2) An *acceptance* mechanism that accepts the proposed  $y$  with probability  $\min(1, P(y)/P(x))$ . In other words, a point  $y$  with larger posterior probability than the current  $x$  is always accepted, and one with lower probability is accepted randomly with probability  $P(y)/P(x)$ .

To see that this satisfies (11.20) note that, for  $x \neq y$ ,

$$\begin{aligned} P(x)\tau(y|x) &= P(x)F(y|x)\min(1, P(y)/P(x)) \\ &= F(y|x)\min(P(x), P(y)) \\ &= F(x|y)\min(P(y), P(x)) \\ &= P(y)\tau(x|y). \end{aligned}$$

Here we used the symmetry of the proposal mechanism to replace  $F(y|x)$  in the second line by  $F(x|y)$  in the third.

### Gibbs sampling

When we have a probabilistic model of many variables, it may often be possible to sample from the distribution obtained by keeping all variables fixed except one, i.e. the conditional distribution. Gibbs sampling exploits this idea. It works by choosing points from the conditional distribution  $P(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)$  for each  $i$ , cycling repeatedly through  $i = 1, \dots, N$ .

To show that this samples correctly from  $P$ , it is enough to prove detailed balance. This means that

$$\begin{aligned} P(x_1, \dots, x_n)P(\tilde{x}_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \\ = P(x_1, \dots, x_{i-1}, \tilde{x}_i, x_{i+1}, \dots, x_n)P(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n). \end{aligned}$$

But we can rewrite this as

$$\begin{aligned} P(x_1, \dots, x_n)P(x_1, \dots, x_{i-1}, \tilde{x}_i, x_{i+1}, \dots, x_n)/P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \\ = P(x_1, \dots, x_{i-1}, \tilde{x}_i, x_{i+1}, \dots, x_n)P(x_1, \dots, x_n)/P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n), \end{aligned}$$

which makes the equality obvious. Provided that the process doesn't get stuck in some subset of the parameter space, i.e. provided it is ergodic, Gibbs sampling will inevitably converge to  $P$ .

The kind of situation in which Gibbs sampling can get stuck is where there are two pieces of density which do not overlap along any of the coordinate directions, e.g. in the 2D case where half the density lies in the region  $[0, 1] \times [0, 1]$  and the other half in the region  $[2, 3] \times [2, 3]$ . Note that if there were even a small overlap, e.g. if half the density were uniform on  $[0, 1] \times [0, 1]$  and the other half uniform on  $[0.99, 1.99] \times [0.99, 1.99]$ , then sampling would pass between the two regions, albeit making the transition between regions quite infrequently.

## Exercise

- 11.13 What is the expected number of samples within one region, in the preceding example, before a cross-over occurs into the other?

## 11.5 Estimation of probabilities from counts

Above we used the example of rolling a die. We needed to estimate the parameters of a multinomial from data: rolls of the die. The same abstract situation occurs frequently in sequence analysis, but with the number of rolls  $n_i$  with outcome  $i$  now meaning something different. For instance,  $n_i$  might be the number of times amino acid  $i$  occurs in a column of a multiple alignment.

Assume that the observations can be expressed as counts  $n_i$  for outcome  $i$  ( $i = 1, \dots, K$ ) and we want to estimate the probabilities  $\theta_i$  for the underlying multinomial distribution. If we have plenty of data, it is natural to use the observed frequencies,  $\theta_i = n_i/N$ , as the estimated probabilities. Here  $N = \sum_i n_i$ . This is the maximum likelihood solution,  $\theta_i^{\text{ML}}$ . The proof that this is so goes as follows.

We want to show that  $P(n|\theta^{\text{ML}}) > P(n|\theta)$  for any  $\theta \neq \theta^{\text{ML}}$ . This is equivalent to showing that  $\log[P(n|\theta^{\text{ML}})/P(n|\theta)] > 0$ , if we only consider probability parameters yielding a non-zero probability. Using equations (11.3) and the definition of  $\theta^{\text{ML}}$ , this becomes

$$\begin{aligned} \log \frac{P(n|\theta^{\text{ML}})}{P(n|\theta)} &= \log \frac{\prod_i (\theta_i^{\text{ML}})^{n_i}}{\prod_i \theta_i^{n_i}} \\ &= \sum_i n_i \log \frac{\theta_i^{\text{ML}}}{\theta_i} \\ &= N \sum_i \theta_i^{\text{ML}} \log \frac{\theta_i^{\text{ML}}}{\theta_i} > 0. \end{aligned}$$

The last inequality follows from the fact that the relative entropy (11.10) is always positive except when the two distributions are identical. This concludes the proof.

If data are scarce, it is not so clear what is the best estimate. If, for instance, we only have a total of two counts both on the same residue, the maximum likelihood estimate would give zero probability to all other residues. In this case, we would like to assign some probability to the other residues and not rely entirely on so few observations. Since there are no more observations, these probabilities must be determined from *prior knowledge*. This can be done via Bayesian statistics, and we will now derive the posterior mean estimator for  $\theta$ .

As the prior we choose the Dirichlet distribution (11.5) with parameters  $\alpha$ . We can then calculate the posterior (11.13) for the multinomial distribution with

observations  $n$ :

$$P(\theta|n) = \frac{P(n|\theta)\mathcal{D}(\theta|\alpha)}{P(n)}.$$

For ease of notation, we have dropped the conditioning on the model  $M$  as compared to (11.13), and consider all probabilities implicitly conditioned on the model. Inserting the multinomial distribution (11.3) for  $P(n|\theta)$  and the expression (11.5) for  $\mathcal{D}(\theta|\alpha)$  yields

$$P(\theta|n) = \frac{1}{P(n)Z(\alpha)M(n)} \prod_i \theta_i^{n_i + \alpha_i - 1} = \frac{Z(n + \alpha)}{P(n)Z(\alpha)M(n)} \mathcal{D}(\theta|n + \alpha).$$

In the last step  $\prod_i \theta_i^{n_i + \alpha_i - 1}$  was recognised as being proportional to the Dirichlet distribution with parameters  $n + \alpha$ . Here  $n + \alpha$  means the set of parameters  $\{n_i + \alpha_i\}$  (vector addition). Fortunately we do not have to get involved with gamma functions in order to finish the calculation, because we know that both  $P(\theta|n)$  and  $\mathcal{D}(\theta|n + \alpha)$  are properly normalised probability distributions over  $\theta$ . This means that all the prefactors must cancel and

$$P(\theta|n) = \mathcal{D}(\theta|n + \alpha). \quad (11.22)$$

We see that the posterior is itself a Dirichlet distribution like the prior, but of course with different parameters. The observation that the above prefactor is one gives us a little corollary, which will be useful later:

$$P(n) = \frac{Z(n + \alpha)}{Z(\alpha)M(n)}. \quad (11.23)$$

Now, we only need to perform an integral in order to find the posterior mean estimator. From the definition (11.15),

$$\theta_i^{\text{PME}} = \int \theta_i \mathcal{D}(\theta|n + \alpha) d\theta = Z^{-1}(n + \alpha) \int \theta_i \prod_k \theta_k^{n_k + \alpha_k - 1} d\theta. \quad (11.24)$$

We can bring  $\theta_i$  inside the product giving  $\theta_i^{n_i + \alpha_i}$  as the  $i$ th term. Then we see that the integral is exactly of the form (11.6). We can therefore write

$$\begin{aligned} \theta_i^{\text{PME}} &= \frac{Z(n + \alpha + \delta_i)}{Z(n + \alpha)} \\ &= \frac{n_i + \alpha_i}{N + A}, \end{aligned} \quad (11.25)$$

where  $A = \sum_i \alpha_i$ , and  $\delta_i$  is a vector whose  $i$ th component is one and all its other components zero. Here we have used the property (11.7) of the gamma function, i.e.  $\Gamma(x + 1) = x\Gamma(x)$ ; this allows us to cancel all terms except  $n_i + \alpha_i$  in the numerator and  $N + A$  in the denominator.

This result should be compared to the ML estimate  $\theta^{\text{ML}}$ . If we think of the  $\alpha$ s as extra observations added to the real ones, this is precisely the ML estimate!

The  $\alpha$ s are like *pseudocounts* added to the real counts. This makes the Dirichlet regulariser very intuitive, and we can in a sense forget all about Bayesian statistics and think in terms of pseudocounts. It is fairly obvious how to use these pseudocounts: if it is known *a priori* that a certain residue, say number  $i$ , is very common, we should give it a high pseudocount  $\alpha_i$ , and if residue  $j$  is generally rare, we should give it a low pseudocount.

It is important to note the self-regulating property of the pseudocount regulariser. If there are many observations, i.e. the  $n$ s are much larger than the  $\alpha$ s, then the estimate is essentially equal to the ML estimate. On the other hand, if there are very few observations, the regulariser would dominate and give an estimate close to the normalised  $\alpha$ s,  $\theta_i \simeq \alpha_i / A$ . So typically we would choose the  $\alpha$ s so that they are equal to the overall distribution of residues after normalisation.

### Mixtures of Dirichlets

It is not easy to express all the prior knowledge about proteins in a single Dirichlet distribution; to achieve that it is natural to use several different Dirichlet distributions. We might for instance have a Dirichlet well suited to exposed amino acids, one for buried ones and so forth. In statistical terms this can be expressed as a *mixture* distribution. Assume we have  $m$  Dirichlet distributions characterised by parameter vectors  $\alpha^1, \dots, \alpha^m$ . A mixture prior expresses the idea that any probability vector  $\theta$  belongs to one of the components of the mixture  $\mathcal{D}(\theta|\alpha^k)$  with a probability  $q_k$ . Formally:

$$P(\theta|\alpha^1, \dots, \alpha^m) = \sum_k q_k \mathcal{D}(\theta|\alpha^k), \quad (11.26)$$

where  $q_k$  are called the mixture coefficients. The mixture coefficients have to be positive and sum to one in order for the mixture to be a proper probability distribution. (Mixtures can be formed from any types of distributions in this way.) Whereas this probability was called  $P(\theta)$  in the previous section, we are here conditioning on the  $\alpha$ s, which was implicit before. This turns out to be convenient, because we can then use probabilities like  $P(\alpha^1|n)$  below. We can then also identify  $q_k$  as the *prior* probability  $q_k = P(\alpha^k)$  of each of the mixture coefficients.

For a given mixture, i.e. for fixed  $\alpha$  parameters and mixture coefficients, it is straightforward to calculate the posterior probabilities using the results from the previous section. From the definition of conditional probabilities, we have

$$\begin{aligned} P(\theta|n) &= \sum_k P(\theta|\alpha^k, n) P(\alpha^k|n) \\ &= \sum_k P(\alpha^k|n) \mathcal{D}(\theta|n + \alpha^k), \end{aligned}$$

where we used the expression for the posterior (11.22). To compute the term

$P(\alpha^k|n)$ , note that by Bayes' theorem we have

$$P(\alpha^k|n) = \frac{q_k P(n|\alpha^k)}{\sum_l q_l P(n|\alpha^l)},$$

using  $q_k = P(\alpha^k)$ . The probability  $P(n|\alpha^k)$  is given by (11.23) (remember that  $P(n)$  in the previous section was implicitly conditioned on the Dirichlet parameters, so it is  $P(n|\alpha^k)$ ), and we get

$$P(\alpha^k|n) = \frac{q_k Z(n + \alpha^k) / Z(\alpha^k)}{\sum_l q_l Z(n + \alpha^l) / Z(\alpha^l)}. \quad (11.27)$$

The final integration to obtain  $\theta^{\text{PME}}$  can be done using the results (11.24) and (11.25) from the previous section, and yields

$$\theta_i^{\text{PME}} = \sum_k P(\alpha^k|n) \frac{n_i + \alpha_i^k}{N + A}. \quad (11.28)$$

The estimate using a mixture of Dirichlets is similar to using a single one: you just average the estimate based on each component of the mixture. However, the weight (11.27) with which they are averaged in the mixture is new. This weight is a little hard to understand intuitively, but it gives a high weight to mixture components with a high probability for the sample.

### Estimating the prior

For more details of the ideas presented in the preceding section, see Brown *et al.* [1993] and Sjölander *et al.* [1996]. These authors used Dirichlet mixtures to model the distribution of column counts. They obtained the prior by estimating the mixture components and the mixture coefficients from a large dataset, i.e. a large set of count vectors.

The estimation is done as follows: The mixture defines a probability for each count vector in the database,  $n^1, \dots, n^M$ ,

$$P(n^l | \alpha^1, \dots, \alpha^m; q_1, \dots, q_m) = \int P(n^l | \theta) P(\theta | \alpha^1, \dots, \alpha^m; q_1, \dots, q_m) d\theta. \quad (11.29)$$

If the count vectors are considered independent, the total likelihood of the mixture is

$$P(\text{data} | \text{mixture}) = \prod_{l=1}^M P(n^l | \alpha^1, \dots, \alpha^m; q_1, \dots, q_m). \quad (11.30)$$

This probability can be maximised by gradient descent or some other method of continuous optimisation.

At this point the reader is probably asking 'Why use ML estimation instead of

these wonderful Bayesian approaches I just learned?' To do this you just need a prior on the parameters of the first level of priors. You can put priors on prior parameters forever. At some point you have to settle for a prior you invented or one estimated by ML or some other non-Bayesian method.

## 11.6 The EM algorithm

The expectation maximisation (EM) algorithm is a general algorithm for ML estimation with 'missing data' [Dempster, Laird & Rubin 1977]. The Baum-Welch algorithm for estimating hidden Markov model probabilities is a special case of the EM algorithm. For HMMs the missing data are the unknown states, since we only know the observations and not the sequence of states producing them.

Assume some statistical model is determined by parameters  $\theta$ . The observed quantities are called  $x$ , and the probability of  $x$  is determined by some missing data  $y$ . For the HMM that we will treat below,  $\theta$  is the set of all model parameters  $a$  and  $e$ , and  $y$  represents the path through the model. The aim is to find the model that maximises the log likelihood

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta).$$

Here and in the following  $x$  means all the observations whether there is one or more sequences. To return to the notation with all sequences shown explicitly requires an extra sum over sequences in all the following formulae.

Assume now that we have a valid model,  $\theta^t$ . We want to estimate a new and better model,  $\theta^{t+1}$ . Using  $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$ , we can write the log likelihood as

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta).$$

Multiplying this with  $P(y|x, \theta^t)$  and summing over  $y$  yields

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta).$$

The first term on the right we will call  $Q(\theta|\theta^t)$ ,

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta). \quad (11.31)$$

We want  $\log P(x|\theta)$  to be larger than  $\log P(x|\theta^t)$ , so the difference should be positive. Using the two equations above we can write the difference

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)}.$$



The last term is the relative entropy (11.10) of  $P(y|x, \theta')$  relative to  $P(y|x, \theta)$ , so it is always non-negative, so

$$\log P(x|\theta) - \log P(x|\theta') \geq Q(\theta|\theta') - Q(\theta'|\theta') \quad (11.32)$$

with equality only if  $\theta = \theta'$ , or if  $P(y|x, \theta') = P(y|x, \theta)$  for some other  $\theta \neq \theta'$ . Choosing

$$\theta^{t+1} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta') \quad (11.33)$$

will always make the difference positive and thus the likelihood of the new model larger than the likelihood of  $\theta'$ . Of course, if a maximum has already been reached,  $\theta^{t+1} = \theta'$ , and the likelihood will not change.

The function  $Q$  in (11.31) is an average of  $\log P(x, y|\theta)$  over the distribution of  $y$  obtained with the current set of parameters  $\theta'$ . This can often be expressed analytically as a function of  $\theta$  in which the constants are expectation values in the old model. This will be more concrete when we go through the derivation for HMMs shortly. The EM algorithm is usually formulated like this:

**Algorithm: Expectation maximisation**

**E-step:** Calculate the  $Q$  function (11.31).

**M-step:** Maximise  $Q(\theta|\theta')$  with respect to  $\theta$ . ◁

We saw above that the likelihood increases in each iteration, so the procedure will always reach a local (or maybe global) maximum asymptotically as  $t \rightarrow \infty$ . For many models, such as HMMs, both of these steps can be carried out analytically. If the second step cannot be carried out exactly, we can use some numerical optimisation technique to maximise  $Q$ . In fact, it is not necessary to maximise it; it is enough to make  $Q(\theta^{t+1}|\theta')$  larger than  $Q(\theta'|\theta')$ . Algorithms that increase  $Q$  – without necessarily maximising it – are called generalised EM (GEM) algorithms [Dempster, Laird & Rubin 1977]. Other generalisations of the EM idea can be found in Meng & Rubin [1992] and Neal & Hinton [1993].

### EM explanation of the Baum–Welch algorithm

For the HMM we shall now sketch the derivation of the EM steps which forms the Baum–Welch algorithm described in Chapter 3, p. 63. In this case we want to maximise the likelihood

$$\log P(x|\theta) = \sum_{\pi} \log P(x, \pi|\theta),$$

so the ‘missing data’ are the state paths  $\pi$ . Then  $Q$  (11.31) is given by

$$Q(\theta|\theta') = \sum_{\pi} P(\pi|x, \theta') \log P(x, \pi|\theta). \quad (11.34)$$

to  $P(y|x, \theta)$ , so

(11.32)

the other  $\theta \neq \theta'$ .

(11.33)

of the new model  
has already been

for the distribution  
often be expressed  
expectation values in  
the derivation for  
is:

so the procedure  
algorithmically as it  
can be carried out  
we can use some  
is not necessary to  
( $\theta'$ ). Algorithms  
and generalised EM  
generalisations of the  
algorithm [1993].

algorithm  
steps which forms  
this case we want to

of the  
algorithm

given by

(11.34)

For a given path each parameter in the model will appear some number of times in  $P(x, \pi|\theta)$  given by the product (3.6). If it is a transition probability we will call this number  $A_{kl}(\pi)$  and for the emission probabilities  $E_k(b, \pi)$ , i.e.  $E_k(b, \pi)$  is the number of times character  $b$  is observed in state  $k$  for path  $\pi$  (it depends on the observation sequence, which we do not show explicitly). Then we can write (3.6) as

$$P(x, \pi|\theta) = \prod_{k=1}^M \prod_b [e_k(b)]^{E_k(b, \pi)} \prod_{k=0}^M \prod_{l=1}^M a_{kl}^{A_{kl}(\pi)},$$

where the first product is over all characters  $b$  in the alphabet. After taking the logarithm, (11.34) can now be written as

$$Q(\theta|\theta') = \sum_{\pi} P(\pi|x, \theta') \times \left[ \sum_{k=1}^M \sum_b E_k(b, \pi) \log e_k(b) + \sum_{k=0}^M \sum_{l=1}^M A_{kl}(\pi) \log a_{kl} \right]. \quad (11.35)$$

We observe that the expected values  $A_{kl}$  and  $E_k(b)$  as defined in (3.20) and (3.21) on p. 64 for the Baum-Welch algorithm can be written as expectations of  $A_{kl}(\pi)$  and  $E_k(b, \pi)$  with respect to  $P(\pi|x, \theta')$ :

$$E_k(b) = \sum_{\pi} P(\pi|x, \theta') E_k(b, \pi) \quad \text{and} \quad A_{kl} = \sum_{\pi} P(\pi|x, \theta') A_{kl}(\pi).$$

Doing the sum over  $\pi$  first in (11.35) therefore gives

$$Q(\theta|\theta') = \sum_{k=1}^M \sum_b E_k(b) \log e_k(b) + \sum_{k=0}^M \sum_{l=1}^M A_{kl} \log a_{kl}. \quad (11.36)$$

Finally, we have to show that (3.18) maximises (11.36). Let us first look at the  $A$  term. The difference between this term for  $a_{ij}^0 = \frac{A_{ij}}{\sum_k A_{ik}}$  and for any other  $a_{ij}$  is

$$\sum_{k=0}^M \sum_{l=1}^M A_{kl} \log \frac{a_{kl}^0}{a_{kl}} = \sum_{k=0}^M \left( \sum_{l'} A_{kl'} \right) \sum_{l=1}^M a_{kl}^0 \log \frac{a_{kl}^0}{a_{kl}}.$$

The last expression is the relative entropy (11.10), and thus it is larger than zero unless  $a_{kl} = a_{kl}^0$ . This proves that the maximum is at  $a_{kl}^0$ . Exactly the same procedure can be used for the  $E$  term.

For the HMM the E-step of the EM algorithm consists of calculating the expectations  $E_k(b)$  and  $A_{kl}$ . This is done by the forward-backward procedure as described in Chapter 3. This completely determines the  $Q$  function, and the maximum is expressed directly in terms of these numbers. Therefore the M-step just consists of plugging  $E_k(b)$  and  $A_{kl}$  into the re-estimation formulae for  $e_k(b)$  and  $a_{kl}$  given in (3.18).

## Bibliography

- Abrahams, J. P., van den Berg, M., van Batenburg, E. and Pleij, C. 1990. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research* 18:3035-3044.
- Allison, L. and Wallace, C. S. 1993. The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimisation of multiple alignments. Technical Report TR 93/188, Monash University Computer Science.
- Allison, L., Wallace, C. S. and Yee, C. N. 1992a. Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution* 35:77-89.
- Allison, L., Wallace, C. S. and Yee, C. N. 1992b. Minimum message length encoding, evolutionary trees and multiple alignment. In *Hawaii International Conference on System Sciences*, volume 1, 663-674.
- Altschul, S. F. 1989. Gap costs for multiple sequence alignment. *Journal of Theoretical Biology* 138:297-309.
- Altschul, S. F. 1991. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology* 219:555-565.
- Altschul, S. F. and Erickson, B. W. 1986. Optimal sequence alignment using affine gap costs. *Bulletin of Mathematical Biology* 48:603-616.
- Altschul, S. F. and Gish, W. 1996. Local alignment statistics. *Methods in Enzymology* 266:460-480.
- Altschul, S. F. and Lipman, D. J. 1989. Trees, stars, and multiple biological sequence alignment. *SIAM Journal of Applied Mathematics* 49:197-209.
- Altschul, S. F., Carroll, R. J. and Lipman, D. J. 1989. Weights for data related by a tree. *Journal of Molecular Biology* 207:647-653.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215:403-410.
- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* 25:3389-3402.
- Asai, K., Hayamizu, S. and Handa, K. 1993. Prediction of protein secondary structure by the hidden Markov model. *Computer Applications in the Biosciences* 9:141-146.
- Asmussen, S. 1987. *Applied Probability and Queues*. Wiley.
- Atteson, K. 1997. The performance of the neighbor-joining method of phylogeny reconstruction. In Mirkin, B., McMorris, F., Roberts, F. and Rzhetsky, A., eds.,

*Mathematical Hierarchies and Biology*. American Mathematical Society. 133–148.

- Bahl, L. R., Brown, P. F., de Souza, P. V. and Mercer, R. L. 1986. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of ICASSP '86*, 49–52.
- Bailey, T. L. and Elkan, C. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In Altman, R., Brutlag, D., Karp, P., Lathrop, R. and Searls, D., eds., *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28–36. AAAI Press.
- Bailey, T. L. and Elkan, C. 1995. The value of prior knowledge in discovering motifs with MEME. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 21–29. AAAI Press.
- Bairoch, A. and Apweiler, R. 1997. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucleic Acids Research* 25:31–36.
- Bairoch, A., Bucher, P. and Hofmann, K. 1997. The PROSITE database, its status in 1997. *Nucleic Acids Research* 25:217–221.
- Baldi, P. and Brunak, S. 1998. *Bioinformatics – The Machine Learning Approach*. MIT Press.
- Baldi, P. and Chauvin, Y. 1994. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation* 6:307–318.
- Baldi, P. and Chauvin, Y. 1995. Protein modeling with hybrid hidden Markov model/neural network architectures. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 39–47. AAAI Press.
- Baldi, P., Brunak, S., Chauvin, Y. and Krogh, A. 1996. Naturally occurring nucleosome positioning signals in human exons. *Journal of Molecular Biology* 263:503–510.
- Baldi, P., Chauvin, Y., Hunkapiller, T. and McClure, M. A. 1994. Hidden Markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences of the USA* 91:1059–1063.
- Bandelt, H.-J. and Dress, A. W. M. 1992. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. *Molecular Phylogenetics and Evolution* 1:242–252.
- Barton, G. J. 1993. An efficient algorithm to locate all locally optimal alignments between two sequences allowing for gaps. *Computer Applications in the Biosciences* 9:729–734.
- Barton, G. J. and Sternberg, M. J. E. 1987. A strategy for the rapid multiple alignment of protein sequences. *Journal of Molecular Biology* 198:327–337.
- Baserga, S. J. and Steitz, J. A. 1993. The diverse world of small ribonucleoproteins. In Gesteland, R. F. and Atkins, J. F., eds., *The RNA World*. Cold Spring Harbor Press. pp. 359–381.
- Bashford, D., Chothia, C. and Lesk, A. M. 1987. Determinants of a protein fold:

- unique features of the globin amino acid sequence. *Journal of Molecular Biology* 196:199-216.
- Baum, L. E. 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3:1-8.
- Bengio, Y., De Mori, R., Flammia, G. and Kompe, R. 1992. Global optimization of a neural network-hidden Markov model hybrid. *IEEE Transactions on Neural Networks* 3:252-259.
- Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag.
- Berger, M. P. and Munson, P. J. 1991. A novel randomized iterative strategy for aligning multiple protein sequences. *Computer Applications in the Biosciences* 7:479-484.
- Binder, K. and Heerman, D. W. 1988. *Monte Carlo Simulation in Statistical Mechanics*. Springer-Verlag.
- Bird, A. 1987. CpG islands as gene markers in the vertebrate nucleus. *Trends in Genetics* 3:342-347.
- Birney, E. and Durbin, R. 1997. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. In Gaasterland, T., Karp, P., Karplus, K., Ouzounis, C., Sander, C. and Valencia, A., eds., *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 56-64. AAAI Press.
- Bishop, M. J. and Thompson, E. A. 1986. Maximum likelihood alignment of DNA sequences. *Journal of Molecular Biology* 190:159-165.
- Borodovsky, M. and McIninch, J. 1993. GENMARK: parallel gene recognition for both DNA strands. *Computers and Chemistry* 17:123-133.
- Borodovsky, M. Y., Sprizhitsky, Y. A., Golovanov, E. I. and Alexandrov, A. A. 1986a. Statistical patterns in the primary structure of the functional regions of the *Escherichia coli* genome. I. Frequency characteristics. *Molekularnaya Biologia* 20:826-833. (English translation).
- Borodovsky, M. Y., Sprizhitsky, Y. A., Golovanov, E. I. and Alexandrov, A. A. 1986b. Statistical patterns in the primary structure of the functional regions of the *Escherichia coli* genome. II. Nonuniform Markov models. *Molekularnaya Biologia* 20:833-840. (English translation).
- Borodovsky, M. Y., Sprizhitsky, Y. A., Golovanov, E. I. and Alexandrov, A. A. 1986c. Statistical patterns in the primary structure of the functional regions of the *Escherichia coli* genome. III. Computer recognition of coding regions. *Molekularnaya Biologia* 20:1144-1150. (English translation).
- Bowie, J. U., Luthy, R. and Eisenberg, D. 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253:164-170.
- Box, G. E. P. and Tiao, G. C. 1992. *Bayesian Inference in Statistical Analysis*. Wiley-Interscience.
- Branden, C. and Tooze, J. 1991. *Introduction to Protein Structure*. Garland.
- Brendel, V., Beckmann, J. S. and Trifonov, E. N. 1986. Linguistics of nucleotide

Bi

Bi

Br

Br

Bu

Bu

Bu

Bu

Car

Car

Car

Car

Cas

Cav

Cecl

ular Biology

statistical  
ities 3:1-8.

ization of a  
n Neural

ogy for  
Biosciences

cal

ends in

guage for  
aasterland,  
eds.,  
tems for

nt of DNA

ognition for

A. A. 1986a.  
of the  
ya Biologia

A. A. 1986b.  
of the  
larnaya

A. A. 1986c.  
of the  
ons.

otein  
ice

alysis.

and.  
nucleotide

sequences: morphology and comparison of vocabularies. *Journal of Biomolecular Structure and Dynamics* 4:11-20.

Brooks, D. R. and McLennan, D. A. 1991. *Phylogeny, Ecology and Behaviour*. University of Chicago Press.

Brown, M. and Wilson, C. 1995. RNA pseudoknot modeling using intersections of stochastic context-free grammars with applications to database search. Unpublished manuscript available from <http://www.cse.ucsc.edu/research/compbio/pseudoknot.html>.

Brown, M., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K. and Haussler, D. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In Hunter, L., Searls, D. B. and Shavlik, J., eds., *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, 47-55. AAAI Press.

Brunak, S., Engelbrecht, J. and Knudsen, S. 1991. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *Journal of Molecular Biology* 220:49-65.

Bucher, P. and Hofmann, K. 1996. A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system. In States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F., eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 44-51. AAAI Press.

Bucher, P., Karplus, K., Moeri, N. and Hofmann, K. 1996. A flexible motif search technique based on generalized profiles. *Computers and Chemistry* 20:3-24.

Buneman, P. 1971. The recovery of trees from measures of dissimilarity. In Hodson, F. R., Kendall, D. G. and Tautu, P., eds., *Mathematics in the Archaeological and Historical Sciences*. Edinburgh University Press. pp. 387-395.

Burge, C. and Karlin, S. 1997. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology* 268:78-94.

Camin, J. H. and Sokal, R. R. 1965. A method for deducing branching sequences in phylogeny. *Evolution* 19:311-327.

Cardon, L. R. and Stormo, G. D. 1992. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *Journal of Molecular Biology* 223:159-170.

Carrillo, H. and Lipman, D. 1988. The multiple sequence alignment problem in biology. *SIAM Journal of Applied Mathematics* 48:1073-1082.

Cary, R. B. and Stormo, G. D. 1995. Graph-theoretic approach to RNA modeling using comparative data. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 75-80. AAAI Press.

Casella, G. and Berger, R. L. 1990. *Statistical Inference*. Duxbury Press.

Cavender, J. A. 1978. Taxonomy with confidence. *Mathematical Biosciences* 40:271-280.

Cech, T. R. and Bass, B. L. 1986. Biological catalysis by RNA. *Annual Review of Biochemistry* 55:599-629.

- Chan, S. C., Wong, A. K. C. and Chiu, D. K. Y. 1992. A survey of multiple sequence comparison methods. *Bulletin of Mathematical Biology* 54:563-598.
- Chang, W. I. and Lawler, E. L. 1990. Approximate string matching in sublinear expected time. In *Proceedings of the 31st Annual IEEE Symposium on Foundations Computer Science*, 116-124. IEEE.
- Chao, K. M., Hardison, R. C. and Miller, W. 1994. Recent developments in linear-space alignment methods: a survey. *Journal of Computational Biology* 1:271-291.
- Chao, K. M., Pearson, W. R. and Miller, W. 1992. Aligning two sequences within a specified diagonal band. *Computer Applications in the Biosciences* 8:481-487.
- Chiu, D. K. Y. and Kolodziejczak, T. 1991. Inferring consensus structure from nucleic acid sequences. *Computer Applications in the Biosciences* 7:347-352.
- Chomsky, N. 1956. Three models for the description of language. *IRE Transactions Information Theory* 2:113-124.
- Chomsky, N. 1959. On certain formal properties of grammars. *Information and Control* 2:137-167.
- Chothia, C. and Lesk, A. M. 1986. The relation between the divergence of sequence and structure in proteins. *EMBO Journal* 5:823-826.
- Churchill, G. A. 1989. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology* 51:79-94.
- Churchill, G. A. 1992. Hidden markov chains and the analysis of genome structure. *Computers and Chemistry* 16:107-115.
- Claverie, J.-M. 1994. Some useful statistical properties of position-weight matrices. *Computers and Chemistry* 18:287-294.
- Collado-Vides, J. 1989. A transformational-grammar approach to the study of the regulation of gene expression. *Journal of Theoretical Biology* 136:403-425.
- Collado-Vides, J. 1991. A syntactic representation of units of genetic information - a syntax of units of genetic information. *Journal of Theoretical Biology* 148:401-429.
- Corpet, F. and Michot, B. 1994. RNAalign program: alignment of RNA sequences using both primary and secondary structures. *Computer Applications in the Biosciences* 10:389-399.
- Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory*. John Wiley & Sons, Inc.
- Cox, D. R. 1962. Further results on tests of separate families of hypotheses. *Journal of the Royal Statistical Society, B* 24:406-424.
- Cox, D. R. and Miller, H. D. 1965. *The Theory of Stochastic Processes*. Chapman & Hall.
- Dandekar, T. and Hentze, M. W. 1995. Finding the hairpin in the haystack: searching for RNA motifs. *Trends in Genetics* 11:45-50.
- Dayhoff, M. O., Eck, R. V. and Park, C. M. 1972. In Dayhoff, M. O., ed., *Atlas of Protein Sequence and Structure*, volume 5. National Biomedical Research Foundation, Washington D.C. pp. 89-99.
- Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C. 1978. A model of evolutionary



- change in proteins. In Dayhoff, M. O., ed., *Atlas of Protein Sequence and Structure*, volume 5, supplement 3. National Biomedical Research Foundation, Washington D.C. pp. 345-352.
- Dembo, A. and Karlin, S. 1991. Strong limit theorems of empirical functionals for large exceedances of partial sums of i.i.d. variables. *Annals of Probability* 19:1737-1755.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* 39:1-38.
- Dong, S. and Searls, D. B. 1994. Gene structure prediction by linguistic methods. *Genomics* 23:540-551.
- Doolittle, R. F., Feng, D.-F., Tsang, S., Cho, G. and Little, E. 1996. Determining divergence times of the major kingdoms of living organisms with a protein clock. *Science* 271:470-477.
- Eck, R. V. and Dayhoff, M. O. 1966. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation.
- Eddy, S. R. 1995. Multiple alignment using hidden Markov models. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 114-120. AAAI Press.
- Eddy, S. R. 1996. Hidden Markov models. *Current Opinion in Structural Biology* 6:361-365.
- Eddy, S. R. and Durbin, R. 1994. RNA sequence analysis using covariance models. *Nucleic Acids Research* 22:2079-2088.
- Eddy, S. R., Mitchison, G. and Durbin, R. 1995. Maximum discrimination hidden Markov models of sequence consensus. *Journal of Computational Biology* 2:9-23.
- Edwards, A. W. F. 1970. Estimation of the branch points of a branching diffusion process. *Journal of the Royal Statistical Society, B* 32:155-174.
- Edwards, A. W. F. 1992. *Likelihood*. Johns Hopkins University Press.
- Edwards, A. W. F. 1996. The origin and early development of the method of minimum evolution for the reconstruction of phylogenetic trees. *Systematic Biology* 45:179-191.
- Edwards, A. W. F. and Cavalli-Sforza, L. 1963. The reconstruction of evolution. *Annals of Human Genetics* 27:105.
- Edwards, A. W. F. and Cavalli-Sforza, L. 1964. Reconstruction of evolutionary trees. In Heywood, V. H. and McNeill, J., eds., *Phenetic and Phylogenetic Classification*. Systematics Association Publication No. 6. pp. 67-76.
- Efron, B. and Tibshirani, R. J. 1993. *An Introduction to the Bootstrap*. Chapman and Hall.
- Efron, B., Halloran, E. and Holmes, S. 1996. Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences of the USA* 93:13429-13434.

- Feller, W. 1971. *An Introduction to Probability Theory and its Applications, Vol II.* John Wiley and Sons.
- Felsenstein, J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics* 25:471-492.
- Felsenstein, J. 1978a. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology* 27:401-410.
- Felsenstein, J. 1978b. The number of evolutionary trees. *Systematic Zoology* 27:27-33.
- Felsenstein, J. 1981a. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* 17:368-376.
- Felsenstein, J. 1981b. A likelihood approach to character weighting and what it tells us about parsimony and compatibility. *Biological Journal of the Linnean Society* 16:183-196.
- Felsenstein, J. 1985. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* 39:783-791.
- Felsenstein, J. 1996. Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Methods in Enzymology* 266:418-427.
- Felsenstein, J. and Churchill, G. A. 1996. A hidden Markov model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution* 13:93-104.
- Feng, D.-F. and Doolittle, R. F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution* 25:351-360.
- Feng, D.-F. and Doolittle, R. F. 1996. Progressive alignment of amino acid sequences and construction of phylogenetic trees from them. *Methods in Enzymology* 266:368-382.
- Fichant, G. A. and Burks, C. 1991. Identifying potential tRNA genes in genomic DNA sequences. *Journal of Molecular Biology* 220:659-671.
- Fields, D. S. and Gutell, R. R. 1996. An analysis of large rRNA sequences folded by a thermodynamic method. *Folding and Design* 1:419-430.
- Fitch, W. M. 1971. Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology* 20:406-416.
- Fitch, W. M. and Margoliash, E. 1967a. Construction of phylogenetic trees. *Science* 155:279-284.
- Fitch, W. M. and Margoliash, E. 1967b. A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as a model case. *Biochemical Genetics* 1:65-71.
- Frasconi, P. and Bengio, Y. 1994. An EM approach to grammatical inference: input/output HMMs. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 2, 289-294. IEEE Comput. Soc. Press.
- Freier, S. M., Kierzek, R., Jaeger, J. A., Sugimoto, N., Caruthers, M. H., Neilson, T. and Turner, D. H. 1986. Improved free-energy parameters for predictions of RNA duplex stability. *Proceedings of the National Academy of Sciences of the USA* 83:9373-9377.
- Fujiwara, Y., Asogawa, M. and Konagaya, A. 1994. Stochastic motif extraction using

hidde  
Searl  
Intell  
Gautheret,  
RNA  
Comy  
Gerstein, M  
accur  
Agar  
Fouri  
59-6  
Gerstein, M  
evolu  
Gersting, J  
Gesteland,  
Labo  
Gilbert, W  
Gold, L., F  
funct  
Goldman,  
Mole  
Goldman,  
prote  
Gonnet, G  
entire  
Gotoh, O.  
of M  
Gotoh, O.  
to m  
9:361  
Gotoh, O.  
by its  
of M  
Grate, L. 1  
conte  
Leng  
Conf  
Gribskov,  
analy  
Gribskov,  
Enzy  
Gribskov,  
dista  
the L

- hidden Markov model. In Altman, R., Brutlag, D., Karp, P., Lathrop, R. and Searls, D., eds., *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 121–129. AAAI Press.
- Gautheret, D., Major, F. and Cedergren, R. 1990. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *Computer Applications in the Biosciences* 6:325–331.
- Gerstein, M. and Levitt, M. 1996. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F., eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 59–67. AAAI Press.
- Gerstein, M., Sonnhammer, E. L. L. and Chothia, C. 1994. Volume changes in protein evolution. *Journal of Molecular Biology* 236:1067–1078.
- Gersting, J. L. 1993. *Mathematical Structures for Computer Science*. W. H. Freeman.
- Gesteland, R. F. and Atkins, J. F., eds. 1993. *The RNA World*. Cold Spring Harbor Laboratory Press.
- Gilbert, W. 1986. The RNA world. *Nature* 319:618.
- Gold, L., Polisky, B., Uhlenbeck, O. and Yarus, M. 1995. Diversity of oligonucleotide functions. *Annual Review of Biochemistry* 64:763–797.
- Goldman, N. 1993. Statistical tests of models of DNA substitution. *Journal of Molecular Evolution* 36:182–198.
- Goldman, N. and Yang, Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution* 11:725–735.
- Gonnet, G. H., Cohen, M. A. and Benner, S. A. 1992. Exhaustive matching of the entire protein sequence database. *Science* 256:1443–1445.
- Gotoh, O. 1982. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* 162:705–708.
- Gotoh, O. 1993. Optimal alignment between groups of sequences and its application to multiple sequence alignment. *Computer Applications in the Biosciences* 9:361–370.
- Gotoh, O. 1996. Significant improvement in accuracy of multiple protein alignments by iterative refinement as assessed by reference to structural alignments. *Journal of Molecular Biology* 264:823–838.
- Grate, L. 1995. Automatic RNA secondary structure determination with stochastic context-free grammars. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 136–144. AAAI Press.
- Gribskov, M. and Veretnik, S. 1996. Identification of sequence patterns with profile analysis. *Methods in Enzymology* 266:198–212.
- Gribskov, M., Lüthy, R. and Eisenberg, D. 1990. Profile analysis. *Methods in Enzymology* 183:146–159.
- Gribskov, M., McLachlan, A. D. and Eisenberg, D. 1987. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences of the USA* 84:4355–4358.

- Gulyaev, A. P. 1991. The computer simulation of RNA folding involving pseudoknot formation. *Nucleic Acids Research* 19:2489-2494.
- Gumbel, E. J. 1958. *Statistics of Extremes*. Columbia University Press.
- Gupta, S. K., Kececioğlu, J. D. and Schaffer, A. A. 1995. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *Journal of Computational Biology* 2:459-472.
- Gutell, R. R. 1993. Collection of small subunit (16S and 16S-like) ribosomal RNA structures. *Nucleic Acids Research* 21:3051-3054.
- Gutell, R. R., Power, A., Hertz, G. Z., Putz, E. J. and Stormo, G. D. 1992. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research* 20:5785-5795.
- Hannenhalli, S., Chappey, C., Koonin, E. V. and Pevsner, P. A. 1995. Genome sequence comparison and scenarios for gene rearrangements: a test case. *Genomics* 30:299-311.
- Harpaz, Y. and Chothia, C. 1994. Many of the immunoglobulin superfamily domains in cell adhesion molecules and surface receptors belong to a new structural set which is close to that containing variable domains. *Journal of Molecular Biology* 238:528-539.
- Harrison, M. A. 1978. *Introduction to Formal Language Theory*. Addison-Wesley.
- Hasegawa, M., Kishino, H. and Yano, T. 1985. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution* 22:160-174.
- Haussler, D., Krogh, A., Mian, I. S. and Sjölander, K. 1993. Protein modeling using hidden Markov models: analysis of globins. In Mudge, T. N., Milutinovic, V. and Hunter, L., eds., *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences*, volume 1, 792-802. IEEE Computer Society Press.
- Hebsgaard, S. M., Korning, P. G., Tolstrup, N., Engelbrecht, J., Rouzé, P. and Brunak, S. 1996. Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information. *Nucleic Acids Research* 24:3439-3452.
- Hein, J. 1989a. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. *Molecular Biology and Evolution* 6:649-668.
- Hein, J. 1989b. A tree reconstruction method that is economical in the number of pairwise comparisons used. *Molecular Biology and Evolution* 6:669-684.
- Hein, J. 1993. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution* 36:396-405.
- Henderson, J., Salzberg, S. and Fasman, K. H. 1997. Finding genes in DNA with a hidden Markov model. *Journal of Computational Biology* 4:127-141.
- Hendy, M. D. and Penny, D. 1989. A framework for the quantitative study of evolutionary trees. *Systematic Zoology* 38:297-309.
- Henikoff, J. G. and Henikoff, S. 1996. Using substitution probabilities to improve

- ig pseudoknot  
position-specific scoring matrices. *Computer Applications in the Biosciences* 12:135–143.
- practical  
pairs multiple  
Henikoff, S. and Henikoff, J. G. 1991. Automated assembly of protein blocks for database searching. *Nucleic Acids Research* 19:6565–6572.
- omal RNA  
Henikoff, S. and Henikoff, J. G. 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the USA* 89:10915–10919.
2. Identifying  
pment and  
ids Research  
Henikoff, S. and Henikoff, J. G. 1994. Position-based sequence weights. *Journal of Molecular Biology* 243:574–578.
- genome  
t case.  
Hertz, G. Z., Hartzell III, G. W. and Stormo, G. D. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computer Applications in the Biosciences* 6:81–92.
- nily domains  
tructural set  
ecular Biology  
Higgins, D. G. and Sharp, P. M. 1989. Fast and sensitive multiple sequence alignments on a microcomputer. *Computer Applications in the Biosciences* 5:151–153.
- on-Wesley.  
plitting by a  
lution  
Higgins, D. G., Bleasby, A. J. and Fuchs, R. 1992. CLUSTAL V: improved software for multiple sequence alignment. *Computer Applications in the Biosciences* 8:189–191.
- deling using  
itinovic, V. and  
nternational  
ter Society  
Hillis, D. M. and Bull, J. J. 1993. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Systematic Biology* 42:182–192.
- and Brunak,  
by combining  
:3439–3452.  
Hillis, D. M., Bull, J. J., White, M. E., Badgett, M. R. and Molineux, I. J. 1992. Experimental phylogenetics: generation of a known phylogeny. *Science* 255:589–592.
- cts ancestral  
ylogeny is  
Hirose, M., Hoshida, M., Ishikawa, M. and Toya, T. 1993. MASCOT: multiple alignment system for protein sequences based on three-way dynamic programming. *Computer Applications in the Biosciences* 9:161–167.
- umber of  
9–684.  
Hirschberg, D. S. 1975. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM* 18:341–343.
- is subject to  
Hogeweg, P. and Hesper, B. 1984. The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of Molecular Evolution* 20:175–186.
- DNA with a  
41.  
Holm, L. and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology* 233:123–138.
- dy of  
Hopcroft, J. E. and Ullman, J. D. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.
- o improve  
Huang, X. and Zhang, J. 1996. Methods for comparing a DNA sequence with a protein sequence. *Computer Applications in the Biosciences* 12:497–506.
- Hudson, R. R. 1990. Gene genealogies and the coalescent process. In Futuyma, D. and Antonovics, J., eds., *Gene Genealogies and the Coalescent Process*. Oxford University Press. pp. 1–44.
- Huelsenbeck, J. P. and Rannala, B. 1997. Phylogenetic methods come of age: testing hypotheses in an evolutionary context. *Science* 276:227–232.
- Hughey, R. and Krogh, A. 1996. Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Computer Applications in the Biosciences* 12:95–107.

- Jacob, F. 1977. Evolution and tinkering. *Science* 196:1161-1166.
- Jefferys, W. H. and Berger, J. O. 1992. Ockham's razor and Bayesian analysis. *American Scientist* 80:64-72.
- Juang, B. H. and Rabiner, L. R. 1991. Hidden Markov models for speech recognition. *Technometrics* 33:251-272.
- Jukes, T. H. and Cantor, C. 1969. Evolution of protein molecules. In *Mammalian Protein Metabolism*. Academic Press. pp. 21-132.
- Karlin, S. and Altschul, S. F. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the USA* 87:2264-2268.
- Karlin, S. and Altschul, S. F. 1993. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proceedings of the National Academy of Sciences of the USA* 90:5873-5877.
- Karplus, K. 1995. Evaluating regularizers for estimating distributions of amino acids. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 188-196. AAAI Press.
- Keeping, E. S. 1995. *Introduction to Statistical Inference*. Dover Publications.
- Kim, J. and Pramanik, S. 1994. An efficient method for multiple sequence alignment. In Altman, R., Brutlag, D., Karp, P., Lathrop, R. and Searls, D., eds., *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 212-218. AAAI Press.
- Kim, J., Pramanik, S. and Chung, M. J. 1994. Multiple sequence alignment using simulated annealing. *Computer Applications in the Biosciences* 10:419-426.
- Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution* 16:111-120.
- Kimura, M. 1983. *The Neutral Theory of Molecular Evolution*. Cambridge University Press.
- Kingman, J. F. C. 1982a. The coalescent. *Stochastic Processes and their Applications* 13:235-248.
- Kingman, J. F. C. 1982b. On the genealogy of large populations. *Journal of Applied Probability* 19A:27-43.
- Kirkpatrick, S., Gelatt, Jr., C. D. and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* 220:671-680.
- Kishino, H., Miyata, T. and Hasegawa, M. 1990. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. *Journal of Molecular Evolution* 31:151-160.
- Konings, D. A. M. and Gutell, R. R. 1995. A comparison of thermodynamic foldings with comparatively derived structures of 16S and 16S-like rRNAs. *RNA* 1:559-574.
- Konings, D. A. M. and Hogeweg, P. 1989. Pattern analysis of RNA secondary structure: similarity and consensus of minimal-energy folding. *Journal of Molecular Biology* 207:597-614.

- analysis.
- ch recognition.
- ammalian
- l significance of  
roceedings of
- ple  
National
- f amino acids.  
id Wodak, S.,  
gent Systems
- ations.
- nce alignment.  
s., *Proceedings  
Molecular*
- nent using  
):419–426.
- f base  
. *Journal of*
- dge University
- ir Applications
- al of Applied
- by simulated
- inference of  
ular Evolution
- amic foldings  
RNA
- ondary  
urnal of
- Krogh, A. 1994. Hidden Markov models for labeled sequences. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 140–144. IEEE Computer Society Press.
- Krogh, A. 1997a. Gene finding: putting the parts together. In Bishop, M., ed., *Guide to Human Genome Computing*. Academic Press, 2nd edition. To appear.
- Krogh, A. 1997b. Two methods for improving performance of a HMM and their application for gene finding. In Gaasterland, T., Karp, P., Karplus, K., Ouzounis, C., Sander, C. and Valencia, A., eds., *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 179–186. AAAI Press.
- Krogh, A. 1998. An introduction to hidden Markov models for biological sequences. In Salzberg, S., Searls, D. and Kasif, S., eds., *Computational Biology: Pattern Analysis and Machine Learning Methods*. Elsevier. Chapter 4. In press.
- Krogh, A. and Mitchison, G. 1995. Maximum entropy weighting of aligned sequences of proteins or DNA. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 215–221. AAAI Press.
- Krogh, A., Mian, I. S. and Haussler, D. 1994. A hidden Markov model that finds genes in *E. coli* DNA. *Nucleic Acids Research* 22:4768–4778.
- Krogh, A., Brown, M., Mian, I. S., Sjölander, K. and Haussler, D. 1994. Hidden Markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology* 235:1501–1531.
- Kuhner, M. K., Yamato, J. and Felsenstein, J. 1995. Estimating effective population size and mutation rate from sequence data using Metropolis–Hastings sampling. *Genetics* 140:1421–1430.
- Kulp, D., Haussler, D., Reese, M. G. and Eeckman, F. H. 1996. A generalized hidden Markov model for the recognition of human genes in DNA. In States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F., eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 134–142. AAAI Press.
- Langley, C. H. and Fitch, W. M. 1974. An examination of the constancy of the rate of molecular evolution. *Journal of Molecular Evolution* 3:161–177.
- Lari, K. and Young, S. J. 1990. The estimation of stochastic context-free grammars using the inside–outside algorithm. *Computer Speech and Language* 4:35–56.
- Lari, K. and Young, S. J. 1991. Applications of stochastic context-free grammars using the inside–outside algorithm. *Computer Speech and Language* 5:237–257.
- Larsen, N. and Zwieb, C. 1993. The signal recognition particle database (SRPDB). *Nucleic Acids Research* 21:3019–3020.
- Law, A. M. and Kelton, W. D. 1991. *Simulation Modelling and Analysis*. McGraw-Hill.
- Lawrence, C. E. and Reilly, A. A. 1990. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins* 7:41–51.
- Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. and Wootton, J. C. 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262:208–214.



- Lefebvre, F. 1995. An optimized parsing algorithm well suited to RNA folding. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 222–230. AAAI Press.
- Lefebvre, F. 1996. A grammar-based unification of several alignment and folding algorithms. In States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F., eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 143–154. AAAI Press.
- Lindenmayer, A. 1968. Mathematical models for cellular interactions in development I. filaments with one-sided inputs. *Journal of Theoretical Biology* 18:280–299.
- Lipman, D. J., Altschul, S. F. and Kececioglu, J. D. 1989. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences of the USA* 86:4412–4415.
- Lisacek, F., Diaz, Y. and Michel, F. 1994. Automatic identification of group I intron cores in genomic DNA sequences. *Journal of Molecular Biology* 235:1206–1217.
- Lowe, T. M. and Eddy, S. R. 1997. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research* 25:955–964.
- Lukashin, A. V., Engelbrecht, J. and Brunak, S. 1992. Multiple alignment using simulated annealing: branch point definition in human mRNA splicing. *Nucleic Acids Research* 20:2511–2516.
- Luthy, R., McLachlan, A. D. and Eisenberg, D. 1991. Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. *Proteins* 10:229–239.
- Luthy, R., Xenarios, I. and Bucher, P. 1994. Improving the sensitivity of the sequence profile method. *Protein Science* 3:139–146.
- MacKay, D. J. C. 1992. Bayesian interpolation. *Neural Computation* 4:415–447.
- MacKay, D. J. C. and Peto, L. 1995. A hierarchical Dirichlet language model. *Natural Language Engineering* 1:1–19.
- Margalit, H., Shapiro, B. A., Oppenheim, A. B. and Maizel, J. V. 1989. Detection of common motifs in RNA secondary structures. *Nucleic Acids Research* 17:4829–4845.
- Mathews, J. and Walker, R. L. 1970. *Mathematical Methods of Physics*. W. A. Benjamin.
- Mau, B., Newton, M. A. and Larget, B. 1996. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Technical Report 961, Statistics Department, University of Wisconsin-Madison.
- Maxwell, E. S. and Fournier, M. J. 1995. The small nucleolar RNAs. *Annual Review of Biochemistry* 64:897–934.
- McCaskill, J. S. 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* 29:1105–1119.
- McClure, M. A., Vasi, T. K. and Fitch, W. M. 1994. Comparative analysis of multiple protein-sequence alignment methods. *Journal of Molecular Evolution* 11:571–592.

folding. In  
Wodak, S.,  
ent Systems

id folding  
.. and Smith,  
Intelligent

i development  
18:280-299.

ltiple sequence  
e USA

roup I intron  
35:1206-1217.

ed detection of  
25:955-964.

nt using  
ing. Nucleic

re-based  
tein sequence

the sequence

415-447.  
nodel. Natural

Detection of  
rch

W. A.

ference via  
ics

nnual Review

inding  
. 119.

is of multiple  
ion

- McKeown, M. 1992. Alternative mRNA splicing. *Annual Review of Cell Biology* 8:133-155.
- Melefors, O. and Hentze, M. W. 1993. Translational regulation by mRNA/protein interactions in eukaryotic cells: ferritin and beyond. *BioEssays* 15:85-90.
- Meng, X.-L. and Rubin, D. B. 1992. Recent extensions to the EM algorithm. *Bayesian Statistics* 4:307-320.
- Mevissen, H. T. and Vingron, M. 1996. Quantifying the local reliability of a sequence alignment. *Protein Engineering* 9:127-132.
- Miller, W. and Myers, E. W. 1988. Sequence comparison with concave weighting functions. *Bulletin of Mathematical Biology* 50:97-120.
- Mitchison, G. 1998. Probabilistic modelling of phylogeny and alignment. *Molecular Biology and Evolution* submitted.
- Mitchison, G. and Durbin, R. 1995. Tree-based maximal likelihood substitution matrices and hidden Markov models. *Journal of Molecular Evolution* 41:1139-1151.
- Miyazawa, S. 1994. A reliable sequence alignment method based on probabilities of residue correspondence. *Protein Engineering* 8:999-1009.
- Mott, R. 1992. Maximum likelihood estimation of the statistical distribution of Smith-Waterman local sequence similarity scores. *Bulletin of Mathematical Biology* 54:59-75.
- Myers, E. W. 1994. A sublinear algorithm for approximate keyword searching. *Algorithmica* 12:345-374.
- Myers, E. W. and Miller, W. 1988. Optimal alignments in linear space. *Computer Applications in the Biosciences* 4:11-17.
- Myers, G. 1995. Approximately matching context-free languages. *Information Processing Letters* 54:85-92.
- Neal, R. M. 1996. *Bayesian Learning in Neural Networks*. Springer (Lecture Notes in Statistics).
- Neal, R. M. and Hinton, G. E. 1993. A new view of the EM algorithm that justifies incremental and other variants. Preprint, Dept. of Computer Science, Univ. of Toronto, available from <ftp://archive.cis.ohio-state.edu/pub/neuroprose/neal.em.ps.Z>.
- Needleman, S. B. and Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48:443-453.
- Noller, H. F., Hoffarth, V. and Zimniak, L. 1992. Unusual resistance of peptidyl transferase to protein extraction procedures. *Science* 256:1416-1419.
- Normandin, Y. and Morgera, S. D. 1991. An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. In *Proceedings of ICASSP '91*, 537-540.
- Nussinov, R., Pieczenik, G., Griggs, J. R. and Kleitman, D. J. 1978. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics* 35:68-82.
- Pavesi, A., Conterlo, F., Bolchi, A., Dieci, G. and Ottonello, S. 1994. Identification of new eukaryotic tRNA genes in genomic DNA databases by a multistep weight

- matrix analysis of transcriptional control regions. *Nucleic Acids Research* 22:1247-1256. F
- Pearson, W. R. 1995. Comparison of methods for searching protein sequence databases. *Protein Science* 4:1145-1160. F
- Pearson, W. R. 1996. Effective protein sequence comparison. *Methods in Enzymology* 266:227-258.
- Pearson, W. R. and Lipman, D. J. 1988. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the USA* 4:2444-2448. S
- Pearson, W. R. and Miller, W. 1992. Dynamic programming algorithms for biological sequence comparison. *Methods in Enzymology* 210:575-601. S
- Pedersen, A. G., Baldi, P., Brunak, S. and Chauvin, Y. 1996. Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. In States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F., eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 182-191. AAAI Press. S
- Peltz, S. W. and Jacobson, A. 1992. mRNA stability: in trans-it. *Current Opinion in Cell Biology* 4:979-983.
- Pesole, G., Attimonelli, M. and Saccone, C. 1994. Linguistic approaches to the analysis of sequence information. *Trends in Biotechnology* 12:401-408. S
- Petrokovski, S., Hirshon, J. and Trifonov, E. N. 1990. Linguistic measure of taxonomic and functional relatedness of nucleotide sequences. *Journal of Biomolecular Structure and Dynamics* 7:1251-1268. S
- Preparata, F. P. and Shamos, M. I. 1985. *Computational Geometry*. Springer-Verlag.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. 1992. *Numerical Recipes in C*. Cambridge University Press. S
- Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77:257-286. S
- Rabiner, L. R. and Juang, B. H. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine* 3:4-16.
- Rabiner, L. R. and Juang, B. H. 1993. *Fundamentals of Speech Recognition*. Prentice-Hall. S
- Rannala, B. and Yang, Z. 1996. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *Journal of Molecular Evolution* 43:304-311. S
- Reese, M. G., Eeckman, F. H., Kulp, D. and Haussler, D. 1997. Improved splice site detection in Genie. *Journal of Computational Biology* 4:311-323.
- Renals, S., Morgan, N., Boulard, H., Cohen, M. and Franco, H. 1994. Connectionist probability estimators in hmm speech recognition. *IEEE Transactions on Speech and Audio Processing* 2:161-174. S
- Riis, S. K. and Krogh, A. 1997. Hidden neural networks: a framework for HMM/NN hybrids. In *Proceedings of ICASSP '97*, 3233-3236. IEEE. S
- Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press. S

- Rosenblueth, D. A., Thieffry, D., Huerta, A. M., Salgado, H. and Collado-Vides, J. 1996. Syntactic recognition of regulatory regions in *Escherichia coli*. *Computer Applications in the Biosciences* 12:415-422.
- Russell, R. B. and Barton, G. J. 1992. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins* 14:309-323.
- Saitou, N. 1996. Reconstruction of gene trees from sequence data. *Methods in Enzymology* 266:427-448.
- Saitou, N. and Nei, M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* 4:406-425.
- Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C. and Haussler, D. 1994. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research* 22:5112-5120.
- Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics* 28:35-42.
- Sankoff, D. and Cedergren, R. J. 1983. Simultaneous comparison of three or more sequences related by a tree. In Sankoff, D. and Kruskal, J. B., eds., *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*. Addison-Wesley. Chapter 9, pp. 253-264.
- Sankoff, D. and Kruskal, J. B. 1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley.
- Sankoff, D., Morel, C. and Cedergren, R. J. 1973. Evolution of 5S RNA and the nonrandomness of base replacement. *Nature New Biology* 245:232-234.
- Schneider, T. D. and Stephens, R. M. 1990. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research* 18:6097-6100.
- Schuster, P. 1995. How to search for RNA structures. Theoretical concepts in evolutionary biotechnology. *Journal of Biotechnology* 41:239-257.
- Schuster, P., Fontana, W., Stadler, P. F. and Hofacker, I. L. 1994. From sequences to shapes and back: a case study in RNA secondary structures. *Proceedings of the Royal Society: Biological Sciences, Series B* 255:279-284.
- Schwartz, R. and Chow, Y.-L. 1990. The N-best algorithm: an efficient and exact procedure for finding the n most likely hypotheses. In *Proceedings of ICASSP'90*, 81-84.
- Searls, D. B. 1992. The linguistics of DNA. *American Scientist* 80:579-591.
- Searls, D. B. and Murphy, K. P. 1995. Automata-theoretic models of mutation and alignment. In Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T. and Wodak, S., eds., *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 341-349. AAAI Press.
- Shapiro, B. A. and Wu, J. C. 1996. An annealing mutation operator in the genetic algorithms for RNA folding. *Computer Applications in the Biosciences* 12:171-180.
- Shapiro, B. A. and Zhang, K. 1990. Comparing multiple RNA secondary structures using tree comparisons. *Computer Applications in the Biosciences* 6:309-318.
- Shimamura, M., Yasue, H., Ohshima, K., Abe, H., Kato, H., Kishiro, T., Goto, M.,



- posons that  
70.
- capiller, T.  
mparison of  
179–191.
- acid sequences  
216:813–818.
- i. and  
ction of weak  
in the
- ular
- ng systematic  
8.
- ehensive  
teins
- ences.
- quences and
- on by  
C. L., eds.,  
1–18. Morgan
- gy
- sites from  
f Sciences of
- different  
al, P.,  
e Fourth  
ogy, 369–375.
- lgorithm of
- lis, D. M. and  
7–511.
- erved  
alignment  
A
- Taylor, W. R. 1987. Multiple sequence alignment by a pairwise algorithm. *Computer Applications in the Biosciences* 3:81–87.
- Thompson, E. A. 1975. *Human Evolutionary Trees*. Cambridge University Press.
- Thompson, J. D., Higgins, D. G. and Gibson, T. J. 1994a. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22:4673–4680.
- Thompson, J. D., Higgins, D. G. and Gibson, T. J. 1994b. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Computer Applications in the Biosciences* 10:19–29.
- Thorne, J. L., Kishino, H. and Felsenstein, J. 1992. Inching toward reality: an improved likelihood model of sequence evolution. *Methods in Enzymology* 34:3–16.
- Tolstrup, N., Rouzé, P. and Brunak, S. 1997. A branch point consensus from Arabidopsis found by non-circular analysis allows for better prediction of acceptor sites. *Nucleic Acids Research* 25:3159–3164.
- Tuerk, C., MacDougall, S. and Gold, L. 1992. RNA pseudoknots that inhibit human immunodeficiency virus type 1 reverse transcriptase. *Proceedings of the National Academy of Sciences of the USA* 89:6988–6992.
- Turner, D. H., Sugimoto, N., Jaeger, J. A., Longfellow, C. E., Freier, S. M. and Kierzek, R. 1987. Improved parameters for prediction of RNA structure. *Cold Spring Harbor Symposia Quantitative Biology* 52:123–133.
- van Batenburg, F. H. D., Gultyaev, A. P. and Pleij, C. W. A. 1995. An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *Journal of Theoretical Biology* 174:269–280.
- Vingron, M. 1996. Near-optimal sequence alignment. *Current Opinion in Structural Biology* 6:346–352.
- Vingron, M. and Waterman, M. S. 1994. Sequence alignment and penalty choice: review of concepts, case studies and implications. *Journal of Molecular Biology* 235:1–12.
- Waterman, M. S. 1995. *Introduction to Computational Biology*. Chapman & Hall.
- Waterman, M. S. and Eggert, M. 1987. A new algorithm for best subsequence alignments with application to tRNA–rRNA comparisons. *Journal of Molecular Biology* 197:723–725.
- Waterman, M. S. and Perlwitz, M. D. 1984. Line geometries for sequence comparisons. *Bulletin of Mathematical Biology* 46:567–577.
- Watson, J. D., Hopkins, N. H., Roberts, J. W., Steitz, J. A. and Weiner, A. M. 1987. *Molecular Biology of the Gene*. Benjamin/Cummings.
- Wilmanns, M. and Eisenberg, D. 1993. Three-dimensional profiles from residue-pair preferences: identification of sequences with beta/alpha-barrel fold. *Proceedings of the National Academy of Sciences of the USA* 90:1379–1383.
- Witherell, G. W., Gott, J. M. and Uhlenbeck, O. C. 1991. Specific interaction between RNA phage coat proteins and RNA. *Progress in Nucleic Acid Research and Molecular Biology* 40:185–220.

- Woese, C. R. and Pace, N. R. 1993. Probing RNA structure, function, and history by comparative analysis. In Gesteland, R. F. and Atkins, J. F., eds., *The RNA World*. Cold Spring Harbor Laboratory Press. pp. 91-117.
- Wray, G. A., Levinto, J. S. and Shapiro, L. H. 1996. Molecular evidence for deep precambrian divergences among metazoan phyla. *Science* 274:568-573.
- Wu, S. and Manber, U. 1992. Fast text searching allowing errors. *Communications of the ACM* 35:83-90.
- Yada, T. and Hirosawa, M. 1996. Detection of short protein coding regions within the Cyanobacterium genome: application of the hidden Markov model. *DNA Research* 3:355-361.
- Yada, T., Sazuka, T. and Hirosawa, M. 1997. Analysis of sequence patterns surrounding the translation initiation sites on Cyanobacterium genome using the hidden Markov model. *DNA Research* 4:1-7.
- Yang, Z. 1993. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution* 10:1396-1401.
- Yang, Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution* 39:306-314.
- Zuckerkandel, E. and Pauling, L. 1962. Molecular disease, evolution and genetic heterogeneity. In Marsha, M. and Pullman, B., eds., *Horizons in Biochemistry*. Academic Press. pp. 189-225.
- Zuker, M. 1989a. Computer prediction of RNA structure. *Methods in Enzymology* 180:262-288.
- Zuker, M. 1989b. On finding all suboptimal foldings of an RNA molecule. *Science* 244:48-52.
- Zuker, M. 1991. Suboptimal sequence alignment in molecular biology: alignment with error analysis. *Journal of Molecular Biology* 221:403-420.
- Zuker, M. and Stiegler, P. 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* 9:133-148.